



Universidad
Carlos III de Madrid

Escuela Politécnica superior
Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**PLATAFORMA PARA LA GESTIÓN ECONÓMICA
Y CIENTÍFICA DE GRUPOS DE INVESTIGACIÓN
BASADA EN TECNOLOGÍAS SEMÁNTICAS**

Autor: Enrique Colen Lozano

Tutor: Juan Miguel Gómez Berbis

AGRADECIMIENTOS

*“Apunta a la luna, incluso si fallas, estarás entre las estrellas”. Gracias a todos los que andáis
el camino a mi lado.*

Resumen

La investigación es la manera más efectiva de generar desarrollo en un país. En nuestro país existe un gran número de grupos que crean trabajo por ámbitos específicos de investigación, permitiendo el desarrollo tanto económico como social de nuestra sociedad.

En la actualidad el uso de los dispositivos móviles, ya sean ordenadores portátiles, teléfonos móviles o tablets, se ha incrementado exponencialmente en los últimos años. Un factor clave para este desarrollo vertiginoso, es la cobertura de internet, la cual entre puntos de acceso wiki y datos móviles, que nos permite tener acceso a internet casi en cualquier lugar.

Por esta razón, hoy en día, una herramienta que pueda ser usada en los distintos dispositivos, suele tener una ventaja importante sobre sus competidores de mercado que no lo hacen.

Por otro lado la información con la que estamos acostumbrados a trabajar a través de internet habitualmente carece de significado semántico. Con la realización de este proyecto se van a aplicar tecnologías semánticas para mejorar una de los puntos débiles que tiene la web actualmente y es que esta carece de contenido semántico.

Con el desarrollo de este proyecto se pretende proporcionar una herramienta web, haciendo uso de las últimas tecnologías para el desarrollo web, como son Spring, bootstrap y mongodb, capaz de gestionar la información de los grupos de investigación desde cualquier dispositivo. Siendo a su vez, una herramienta innovadora, ya que incorporemos tecnologías semánticas para gestionar parte de la información de los proyectos.

Abstract

Researching is the most effective method for the improvement of any country. In our country exists a huge number of groups which generate work in specific researching fields, allowing the economic and social development of our society.

At the moment, the use of mobile devices, even if they are laptops, mobile phones or tablets, has increased exponentially during the past years. A key factor of this vertiginous growth is the Internet coverage, which, among wiki Hotspots and mobile data, allows us to have Internet access in almost every place.

For this reason, at the present day, any tool which can be used in several devices usually has an important advantage over its market competitors that does not have it.

On the other hand, the information we are used to work with over the Internet typically is lacked of semantic meaning. By making this project several semantic technologies will be applied to improve one of the current web's weaknesses: the lack of semantic content.

With the development of this project we are expecting to provide a web tool, making use of the latest technologies for web development, such as Spring, Bootstrap and MongoDB, which are able of manage the information of the researching groups from any device. Being at the same time, an innovative tool, because we are including semantic technologies for managing part of the project's information.

Contenido

ÍNDICE DE TABLAS	7
ÍNDICE DE ILUSTRACIONES.....	9
1 INTRODUCCIÓN	10
1.1 Objetivos	10
1.2 Contenido del documento	10
1.3 Terminología	11
1.3.1 Acrónimos y abreviaturas.....	11
1.4.3 Definiciones.....	12
2 ESTADO DEL ARTE	13
2.1 Grupos de investigación	13
2.1.1 Consejo Superior de Investigaciones Científicas.....	14
2.2 Web Semántica	17
2.2.1 Web actualmente.....	17
2.2.2 Web semántica, la web del futuro	19
2.2.3 Tecnología de la web semántica	21
2.6 Análisis de metodologías para realizar el proyecto	26
2.6.1 Metodologías ágiles	26
2.6.2 Metodología tradicional.....	28
2.6.3 Comparativa entre metodologías ágiles y tradicionales	34
2.7 Solución propuesta.....	35
2.7.1 Herramientas y tecnologías.....	35
2.7.2 Repositorio del proyecto.....	42
2.7.3 Metodología de desarrollo.....	42
3 ANÁLISIS	44
3.1 Descripción general.....	44
3.2 Requisitos de usuario	44
3.3.1 Requisitos de capacidad	45
3.3.2 Requisitos de restricción	48
3.3 Casos de uso.....	49

3.3.1 Diagramas UML	49
3.3.2 Descripción contextual.....	51
3.4 Requisitos de software.....	62
3.4.1 Requisitos de software funcionales	63
3.4.2 Requisitos de software no funcionales	67
3.4.2.1 Requisitos no funcionales de Operación.....	67
3.4.2.2 Requisitos no funcionales de interfaz	67
4 DISEÑO	69
4.1 Arquitectura del sistema	69
4.2 Interfaz web	73
4.2.1 Login	73
4.2.2 Formularios	74
4.2.1 Buscador.....	78
5 PRUEBAS.....	79
5.1 Pruebas de aceptación	79
5.2 Resultados de las pruebas.....	86
6 GESTION DEL PROYECTO	87
6.1 Planificación del proyecto	87
6.2 Análisis económico	88
6.2.1 Gastos de personal.....	88
6.2.2 Gastos Hardware	89
6.2.3 Gastos Software	89
6.2.4 Coste total del proyecto	89
7 CONCLUSIONES	90
7.1 Conclusiones del proyecto	90
7.2 Líneas futuras	91
8 REFERENCIAS y BIBLIOGRAFÍA.....	92

ÍNDICE DE TABLAS

Tabla 1: Tipos RDF	24
Tabla 2: Comparativa metodologías ágiles y tradicionales	34
Tabla 3: Patrón requisito de usuario	44
Tabla 4: RUC-01, Acceso a la aplicación web	45
Tabla 5: RUC-02 , Visualizar Proyectos Asignados	45
Tabla 6: RUC-03, Visualizar un proyecto	46
Tabla 7: RUC-04, Visualizar información del proyecto	46
Tabla 8: RUC-05, Gestionar gastos	46
Tabla 9: RUC-06, Consultar gastos	46
Tabla 10: RUC-07, Añadir Gastos	46
Tabla 11: RUC-08, Gestionar conferencias.....	47
Tabla 12: RUC-09, Consultar conferencias.....	47
Tabla 13: RUC-10, Añadir conferencias.....	47
Tabla 14: RUC-11, Buscar un proyecto.....	47
Tabla 15: RUR-01, Idioma.....	48
Tabla 16: RUR-02, Disponibilidad en varios navegadores.....	48
Tabla 17: RUR-03, Disponibilidad horaria	48
Tabla 18: RUR-04, Interfaz sencilla.....	48
Tabla 19: Plantilla casos de uso.....	51
Tabla 20: CUA-01, Dar de alta un investigador	51
Tabla 21: CUA-02, Dar de baja un investigador	52
Tabla 22: CUA-03, Dar de alta un Proyecto.....	52
Tabla 23: CUI-01, Visualizar proyectos.....	53
Tabla 24: CUI-02, Ver información del proyecto.....	53
Tabla 25: CUI-03, Visualizar gastos.	54
Tabla 26: CUI-04, Consultar gastos.	54
Tabla 27: CUI-05, Añadir gastos.	55
Tabla 28: CUI-06, Visualizar conferencias.	55
Tabla 29: CUI-07, Consultar mis conferencias.....	56
Tabla 30: CUI-08, Buscar proyectos.	56
Tabla 31: CUJ-01, Visualizar proyectos.....	57
Tabla 32: CUJ-02, Ver información del proyecto.	57
Tabla 33: CUJ-03, Añadir investigador.	58
Tabla 34: CUJ-04, Visualizar gastos.	58
Tabla 35: CUJ-05, Consultar gastos.	59
Tabla 36: CUJ-06, Añadir gastos.	59
Tabla 37: CUJ-07, Visualizar conferencias.	60
Tabla 38: CUJ-08, Consultar mis conferencias.	60
Tabla 39: CUJ-09, Añadir conferencias.....	61
Tabla 40: CUJ-10, Buscar proyectos.	61
Tabla 41: Plantilla Requisitos Software	62
Tabla 42: RSF-01, Acceso a la aplicación web usuario administrador.	63
Tabla 43: RSF-02, Acceso a la aplicación web usuario investigador.	63

Tabla 44: RSF-03, Acceso a la aplicación web usuario jefe proyecto.....	63
Tabla 45: RSF-04, Visualizar Proyectos Asignados	63
Tabla 46: RSF-05, Visualizar un proyecto	64
Tabla 47: RSF-06, Visualizar información del proyecto.....	64
Tabla 48: RSF-07, Gestionar gastos	64
Tabla 49: RSF-08, Consultar gastos	64
Tabla 50: RSF-10, Añadir Gastos	65
Tabla 51: RSF-11, Gestionar conferencias.....	65
Tabla 52: RSF-12, Consultar conferencias	65
Tabla 53: RSF-13, Añadir conferencias.....	65
Tabla 54: RSF-14, Buscar un proyecto.....	66
Tabla 55: RSNFO-01, Disponibilidad horaria	67
Tabla 56: RSNFO-02, Disponibilidad en varios navegadores.....	67
Tabla 57: RSNFI-01, Idioma	67
Tabla 58: RSNFI-02, Interfaz sencilla	68
Tabla 59: RSNFI-03, Resolución de la interfaz.....	68
Tabla 60: Patron Pruebas	79
Tabla 61: PA-01, Dar de alta un investigador.....	80
Tabla 62: PA-02, Dar de baja un investigador	80
Tabla 63: PA-03, Dar de alta un proyecto	80
Tabla 64: PA-04, Visualizar proyectos usuario investigador	80
Tabla 65: PA-05, Visualizar proyectos usuario jefe proyecto.....	81
Tabla 66: PA-06, información del proyecto usuario investigador.....	81
Tabla 67: PA-07, Ver información del proyecto usuario jefe proyecto.....	81
Tabla 68: PA-08, Visualizar proyectos usuario investigador	81
Tabla 69: PA-09, Visualizar gastos usuario investigador	82
Tabla 70: PA-10, Visualizar gastos usuario jefe proyecto.	82
Tabla 71: PA-11, Consultar gastos usuario investigador	82
Tabla 72: PA-12, Consultar gastos usuario jefe proyecto.	82
Tabla 73: PA-13, Añadir gastos usuario investigador.....	83
Tabla 74: PA-14, Añadir gastos usuario jefe proyecto.	83
Tabla 75: PA-15, Visualizar conferencias usuario investigador.....	83
Tabla 76: PA-16, Visualizar conferencias usuario jefe proyecto.	84
Tabla 77: PA-17, Consultar conferencias usuario investigador.....	84
Tabla 78: PA-18, Consultar conferencias usuario jefe proyecto.	84
Tabla 79: PA-19, Añadir conferencias usuario jefe proyecto.....	85
Tabla 80: PA-20, Buscar proyectos usuario investigador.....	85
Tabla 81: PA-21, Buscar proyectos usuario jefe proyecto	85
Tabla 82. Resultados pruebas	86
Tabla 83: gastos personal.....	88
Tabla 84: Gastos hardware.....	89
Tabla 85: Gastos software.....	89
Tabla 86: Coste total proyecto	89

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Actividad científico-técnica CSIC 2013	15
Ilustración 2: web semántica vs actual	19
Ilustración 3: Estructura de la web semántica	21
Ilustración 4: Estructura IRI	22
Ilustración 5: Expresión RDF.....	23
Ilustración 6: RUP estructura dinámica.....	31
Ilustración 7: Fases del método en cascada.....	32
Ilustración 8: Bootstrap.....	35
Ilustración 9: MongoDB.....	36
Ilustración 10: Escalabilidad + Velocidad MongoDB	37
Ilustración 11: Spring.....	38
Ilustración 12: Motor Spring	40
Ilustración 13: Apache Marmotta	41
Ilustración 14: GitHub.....	42
Ilustración 15: Diagrama administrador	49
Ilustración 16: Diagrama Investigador	50
Ilustración 17: Diagrama Jefe de proyecto.....	50
Ilustración 18: MVC	69
Ilustración 19: Schema Proyecto.....	71
Ilustración 20: Schema Investigador-Proyectos.....	72
Ilustración 21: Login	73
Ilustración 22: Formulario dar de alta un investigador.....	74
Ilustración 23: Formulario dar de baja un investigador.....	74
Ilustración 24: Formulario dar de Alta un proyecto.....	75
Ilustración 25: Formulario añadir investigador.....	76
Ilustración 26: Formulario añadir gasto	76
Ilustración 27: Formulario añadir conferencia.....	77
Ilustración 28: Buscador.....	78
Ilustración 29: Diagrama Gantt	87

1 INTRODUCCIÓN

El objetivo principal de este apartado es exponer la motivación que ha llevado al desarrollo de este proyecto y los objetivos que se pretenden conseguir con su desarrollo.

1.1 Objetivos

El objetivo principal de este proyecto consiste en diseñar una aplicación web para gestionar los grupos de investigación mediante tecnologías semánticas.

Además esta aplicación deberá ser adaptable a cualquier dispositivo electrónico con acceso a internet.

Con el desarrollo de esta aplicación se permite que un usuario pueda gestionar la información de sus proyectos asignados, así como los gastos y conferencias relativos a estos proyectos.

Investigar y aplicar las tecnologías semánticas disponibles, para mejorar la funcionalidad de nuestra aplicación.

1.2 Contenido del documento

Para facilitar la lectura del documento, a continuación se ofrece una breve descripción de los apartados que componen el documento:

1. Introducción
2. Estado del arte
3. Análisis
4. Diseño
5. Pruebas
6. Gestión del proyecto
7. Conclusiones
8. Biografía

1.3 Terminología

Este apartado consta con un glosario de términos y definiciones que facilitan la comprensión del documento al lector.

1.3.1 Acrónimos y abreviaturas

C.SIC: Consejo superior de investigaciones científicas.

W3C: World Wide Web Consortium.

HTML: HyperText Markup Language.

OWL: Ontology Web Language.

IRI: Internationalized Resource Identifier.

URI: Uniform Resource Identifier.

UCS: Universal character set

URL: Uniform Resource Locator.

URN: Uniform Resource Name.

XML: Extensible markup Language.

CSS: Cascading Style Sheets.

EJB: Enterprise JavaBeans.

IoC: inversión de control.

JNDI: Java Naming and Directory Interface.

MVC: Model View Controller.

IVA: Impuesto sobre el Valor Añadido.

IRPF: Impuesto sobre la Renta de las Personas Físicas.

SS: Seguridad Social.

1.4.3 Definiciones

Meta-datos: Literalmente «sobre datos», son datos que describen otros datos. En general, un grupo de metadatos se refiere a un grupo de datos, llamado recurso.

Algoritmo: Conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad.

Buscador: Sistema informático que busca archivos almacenados

Ontología: Formulación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados.

Parsers: Un parser o procesador o analizador sintáctico lee el documento XML y verifica que es XML bien formado, algunos también comprueban que el código XML sea válido.

JavaScript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Framework: Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

Font icons: Iconos utilizados mediante fuentes tipográficas, ya sean letras y números o caracteres Unicode.

Github: Es una forja para alojar proyectos utilizando el sistema de control de versiones Git.

NoSQL: Es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico del sistema de gestión de bases de datos relacionales.

Wiki: Nombre que recibe un sitio web cuyas páginas pueden ser editadas directamente desde el navegador, donde los usuarios crean, modifican o eliminan contenidos que, generalmente, comparten.

Feedback: Palabra del inglés que significa retroalimentación. Es la reacción, respuesta u opinión que nos da un interlocutor como retorno sobre un asunto determinado.

2 ESTADO DEL ARTE

2.1 Grupos de investigación

Un grupo de investigación es una unidad formada por investigadores con líneas de trabajo unidisciplinares o pluridisciplinares e intereses comunes, que realizan actividades de investigación, desarrollo e innovación, en solitario o en colaboración con otros grupos o entidades, vinculadas o no a la universidad.

En las universidades españolas cada departamento universitario suele tener uno o más grupos de investigación coordinados por un organismo común: los Vicerrectorados de investigación.

Estos organismos se encargan de proporcionar apoyo a los investigadores, además de fomentar el desarrollo de la investigación a través de acciones como la distribución de ayudas destinadas a los distintos grupos investigadores; el estudio, la evaluación y la información de todos los contratos y convenios relacionados con las actividades de investigación.

Los grupos de investigación se encuentran organizados por áreas de conocimiento (Humanidades y Ciencias Sociales, Biología y Biomedicina...) y a su vez se subdividen en otras especialidades más específicas, las cuales dependen del área de conocimiento.

Por otra parte, existen organismos autonómicos, los cuales se encargan de fomentar la innovación y el desarrollo de la investigación. Estos organismos cuentan con buscadores que clasifican los grupos de investigación por área de conocimiento y organismos. En el caso de la Comunidad de Madrid, en su página web recoge los grupos de investigación de las universidades y centros públicos de investigación de esta comunidad. Actualmente existen 4861 grupos de investigación dados de alta en su base de datos.

A nivel nacional existe la agencia estatal del CSIC (consejo superior de investigaciones científicas) que da difusión a la investigación que se desarrolla desde cualquier organismo o centro de nuestro país.

2.1.1 Consejo Superior de Investigaciones Científicas

La Agencia Estatal, Consejo Superior de Investigaciones Científicas (CSIC) es la mayor institución pública dedicada a la investigación en España y la tercera de Europa.

Adscrita al Ministerio de Economía y Competitividad, a través de la Secretaría de Estado de Investigación, Desarrollo e Innovación, su objetivo fundamental es desarrollar y promover investigaciones en beneficio del progreso científico y tecnológico, para lo cual está abierta a la colaboración con entidades españolas y extranjeras.

Acorde con su Estatuto (artículo 4), tiene como misión el fomento, coordinación, desarrollo y difusión de la investigación científica y tecnológica, de carácter pluridisciplinar, con el fin de contribuir al avance del conocimiento y al desarrollo económico, social y cultural, así como a la formación de personal y al asesoramiento de entidades públicas y privadas en esta materia.

El CSIC desempeña un papel central en la política científica y tecnológica de nuestro país, ya que abarca desde la investigación básica a la transferencia del conocimiento al sector productivo.

El motor de la investigación lo forman sus centros e institutos, distribuidos por todas las comunidades autónomas, y sus más de 15.000 trabajadores, de los cuales más de 3.000 son investigadores en plantilla y otros tantos doctores y científicos en formación. El CSIC cuenta con el 6 por ciento del personal dedicado a la Investigación y el Desarrollo en España, que genera aproximadamente el 20 por ciento de la producción científica nacional. Además, gestiona un conjunto de importantes infraestructuras, la red más completa y extensa de bibliotecas especializadas y cuenta con unidades mixtas de investigación.

Sus principales Funciones son:

- Investigación científica y técnica de carácter multidisciplinar
- Asesoramiento científico y técnico
- Transferencia de resultados al sector empresarial
- Contribución a la creación de empresas de base tecnológica
- Formación de personal especializado
- Gestión de infraestructuras y grandes instalaciones
- Fomento de la cultura de la Ciencia
- Representación científica de España en el ámbito internacional

En la siguiente figura se pueden observar los datos del año 2013 (Los últimos que están publicados), de la actividad científico-técnica de Centros e Institutos del CSIC agrupados por áreas de conocimiento.

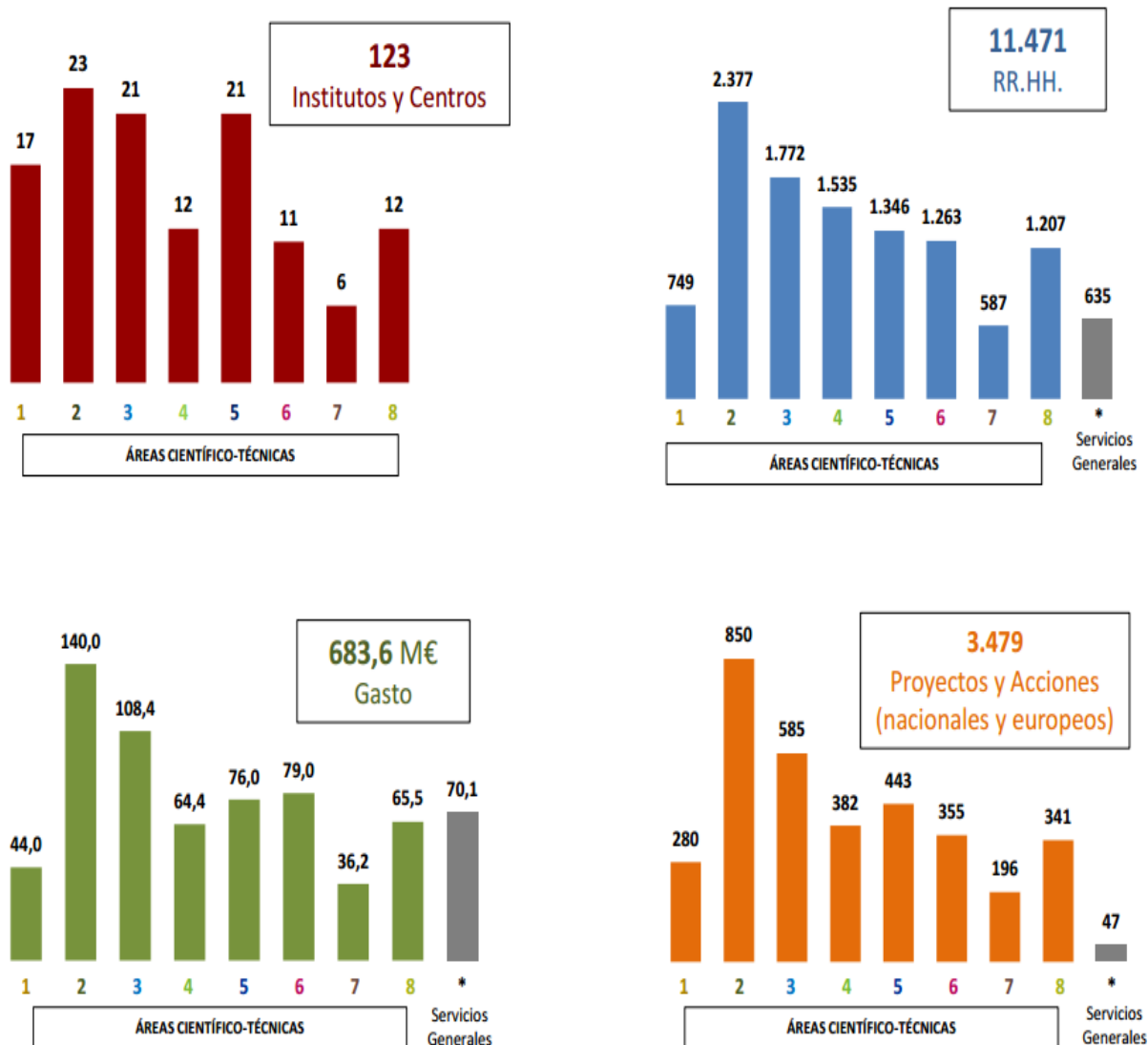


Ilustración 1: Actividad científico-técnica CSIC 2013

Dónde:

- 1 Humanidades y Ciencias Sociales
- 2 Biología y Biomedicina
- 3 Recurso Naturales
- 4 Ciencias Agrarias
- 5 Ciencia y Tecnología Físicas
- 6 Ciencia y Tecnología de Materiales
- 7 Ciencia y Tecnología de Alimentos
- 8 Ciencia y Tecnología Químicas

2.2 Web Semántica

La Web Semántica es un sistema de información basado en Internet en el cual los datos se tratan para ser legibles por los ordenadores. Esta información, gracias a un tratamiento basado en meta-datos, obtiene un significado para las máquinas, permitiéndolas realizar búsquedas con el conocimiento que lo realizaría un humano.

W3C, que es la responsable de la estandarización de la web semántica, la define como:

“La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante.”

2.2.1 Web actualmente

Antes de seguir con la explicación sobre la web semántica, es necesario hacer una parada para ponernos en contexto y analizar la web que tenemos hoy en día.

Actualmente el número de sitios web ha superado los mil millones. El número de sitios web creados al año sigue aumentando cada año. El número de usuarios en YouTube supera los mil millones. Se suben a YouTube más de 300 horas de video por minuto y se reproducen al día cientos de millones de horas de video.

En Facebook el número de usuarios activos es de 936 millones, estos usuarios están una media de 21 minutos dentro de la red social consumiendo contenidos. En twitter existen más de 270 millones de usuarios activos y cada día se envían 500 millones de tweets. Instagram cuenta con 300 millones de usuarios activos, se suben 26 fotografías por segundo, unos 70 millones de fotos nuevas cada día.

Como se puede observar existe una gran cantidad de información almacenada en la web, se puede decir que poco a poco se está convirtiendo en una enciclopedia universal que engloba el conocimiento humano.

Hoy en día la web ofrece multitud de servicios que simplifican o sustituyen las tareas de nuestra vida cotidiana, ya que nos aportan comodidad, eficiencia y facilidad. Se puede leer el periódico, comprar entradas para el cine o un concierto, incluso gestionar nuestras cuentas bancarias, todo desde nuestra casa y al alcance de unos clic del ratón.

Sin embargo la gran cantidad de información que aparece en la web puede constituir un problema a la hora de realizar búsquedas de calidad, escoger el lugar donde pasar las vacaciones, puede resultar una tarea tediosa a pesar de la potencia de los algoritmos de los motores de búsqueda. El buscador ofrece una multitud de resultados al usuario, mientras que este debe navegar por los diferentes sitios para obtener un resultado de calidad. Actualmente la web está pensada para ser entendida por las personas, por lo que desarrollar programas que puedan realizar estas tareas en nuestro lugar es complicado. Es muy difícil reproducir y más costoso aun mantener, en una máquina, la capacidad para comprender los contenidos web que tiene una persona.

Los contenidos web están codificados con el lenguaje HTML. Es un lenguaje de etiquetado de formato que nos permite ordenar la información, pero que carece de significado, por lo que las maquinas no pueden entenderla. Por este motivo los buscadores no pueden distinguir una palabra que sea sinónimo de otra, o que esta palabra tenga más de un significado.

El software ofrece datos a las personas pero sin conocer el significado de la información. La web actual maneja una semántica basada en el peso de las palabras clave, siendo estos significados planos e inconexos. Debido a estos problemas hay pocas posibilidades de conseguir automatizar las tareas que ejecutan los humanos, para que puedan ser reproducidas por una máquina.

2.2.2 Web semántica, la web del futuro

Con la idea de superar las limitaciones de la web actual nace la web semántica. La web semántica propone introducir descriptores explícitos que otorguen significados a los datos. En la web actual predomina la semántica implícita, la ausencia de una organización clara y el crecimiento caótico de recursos, mientras que la web del futuro apuesta por clasificar y dotar de una estructura clara a la nueva información, permitiendo así que esta pueda ser procesada por máquinas.

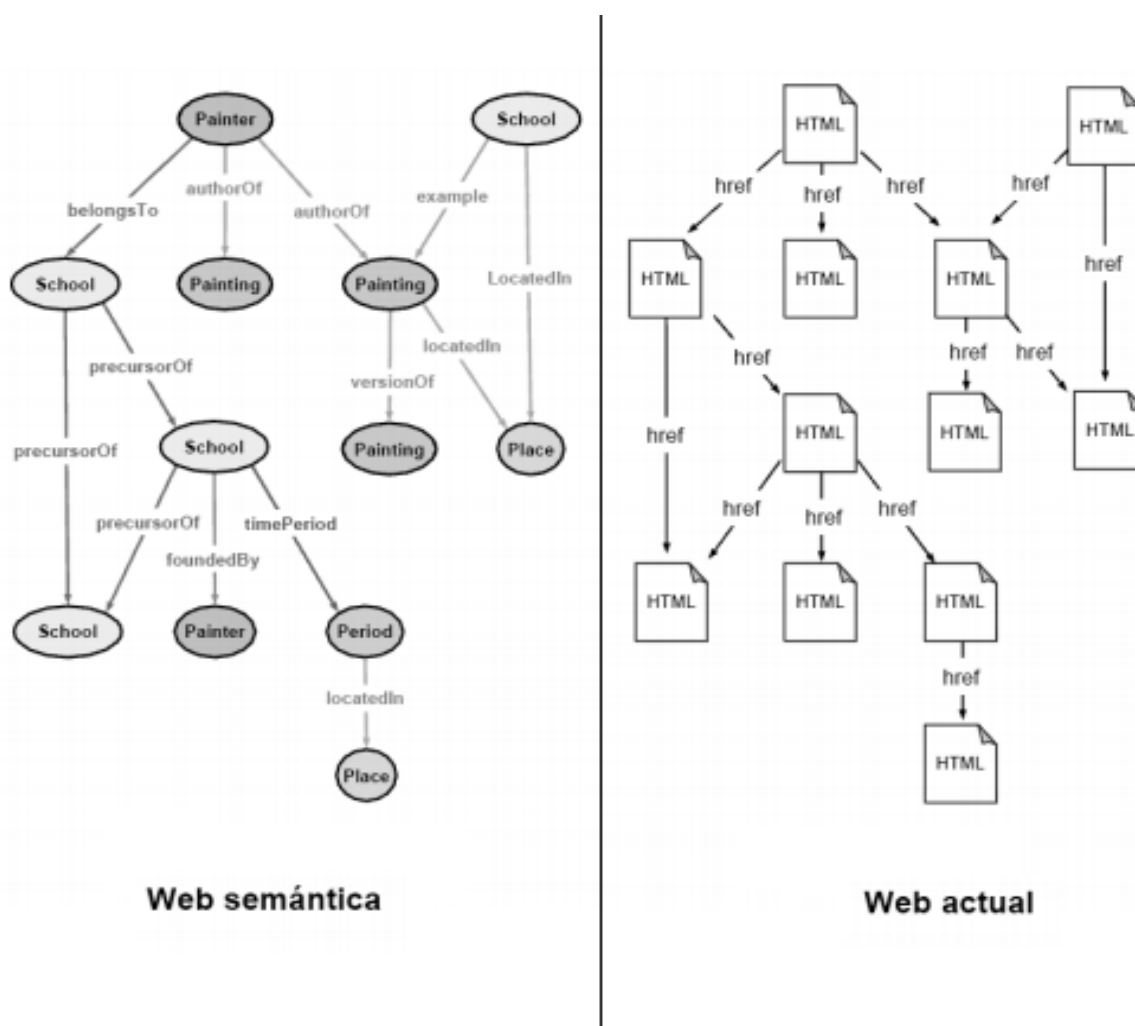


Ilustración 2: web semántica vs actual

En la **Figura anterior** se representa la web desde el punto de vista de una máquina. Podemos observar que en la web actual los enlaces carecen de sentido, simplemente enlazan los documentos entre sí.

Sin embargo la web semántica, se puede observar como cada nodo o recurso y cada enlace posee un significado para la máquina. Observamos los recursos painter (pintor), school (colegio), painting (pintura)... y los enlaces belongs to (pertenece a), autor of (autor de), etc...

Se está creando una web con una estructura de base de datos, donde las consultas realizadas van a ser mucho eficientes. Podremos desde generar la biografía de un autor recopilando la información de los diferentes sitios web, hasta consultar dos los cuadros que ha realizado ese pintor.

Además de los datos y el acceso a información, la web semántica puede influir en otro campo de estudio, como son los servicios web. Realizar Compras de libros, sacar entradas para el cine o conciertos pueden resultar beneficiados por el alcance de este sistema.

Para finalizar vamos a mencionar, las barreras actuales para la imposición de la web semántica. En primer lugar nos encontramos con la falta de un componente tecnológico que transforme la gran cantidad de información existente a un formato que sea comprensible para las máquinas, como es el OWL. La segunda barrera es el modelo de negocio. Muchos de los portales web actuales basan sus ingresos en la publicidad, lo que se traduce en que basas sus ingresos en las visitas obtenidas diariamente. Si toda la información estuviera dotada de contenido semántico la afluencia de usuarios se monopolizaría en aplicaciones capaces de generar webs fruto de recopilaciones de páginas en la red. Por tanto el modelo de negocio no sería sostenible, se debería conseguir modificar un modelo de negocio que genera miles de millones, cosa probable en la sociedad en la que vivimos.

2.2.3 Tecnología de la web semántica

Uno de los principales inconvenientes de la web semántica es la cantidad de tecnologías necesarias que hace falta comprender para poder hacer uso de ella con calidad. Existen lenguajes para la representación de ontologías, parsers, lenguajes de consulta, entornos de desarrollo, módulos de gestión (almacenamiento, acceso, actualización) de ontologías, módulos de visualización, y otras herramientas y librerías.

La siguiente figura muestra una estructura de las tecnologías de la web semántica, que serán descritas a continuación.

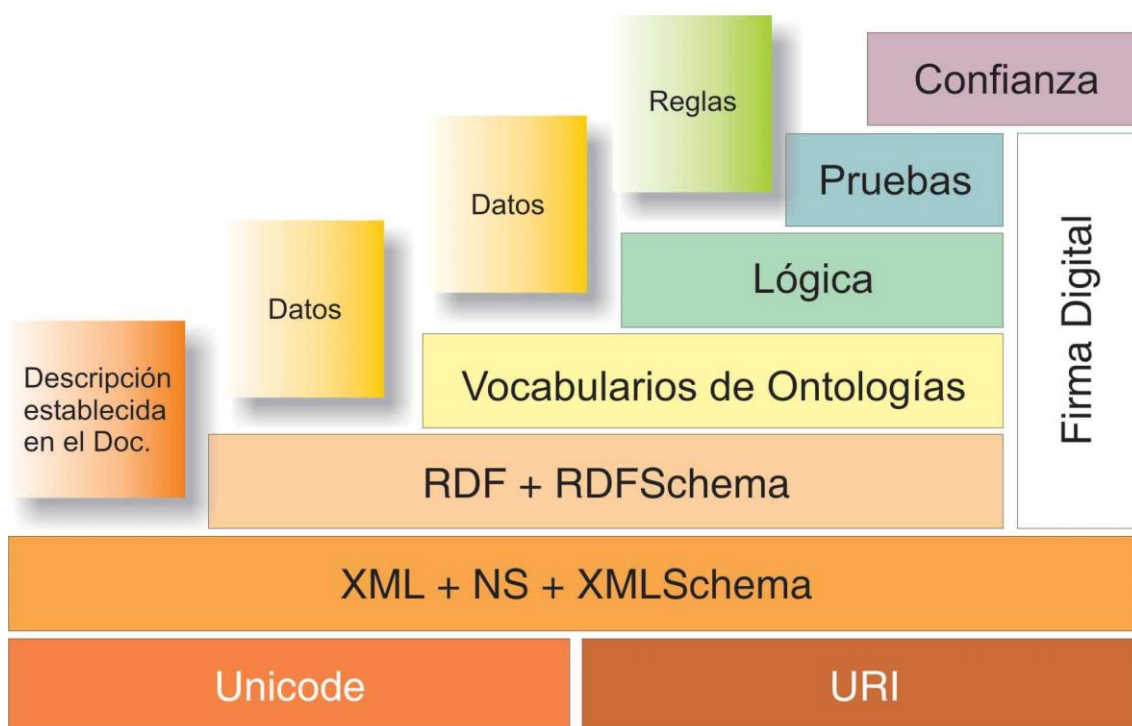


Ilustración 3: Estructura de la web semántica

Tecnologías relacionadas con el hipertexto

Unicode

Codificación de caracteres que debe ser usada para escribir cualquiera de los lenguajes superiores. Es un estándar de codificación para la representación y manipulación del texto que proporciona un número único para cada carácter, sin importar el idioma, el programado la plataforma.

Unicode proporciona un formato estándar de representación de caracteres, creado por el consorcio Unicode, para la creación de cualquier tipo de documento escrito en un computador.

El W3C exige que la web semántica sea escrita con este tipo de codificación.

IRI/URI

IRI (Internationalized Resource Identifier) es una generalización de **URI** (Uniform Resource Identifier), ya que permite usar el Universal character set (**UCS**) mientras que URI solo permite el juego de caracteres americano, a su vez URI es una composición entre un **URL** y una **URN** (Uniform Resource Name) como muestra la siguiente figura. Cualquiera de estos dos elementos puede no aparecer y seguir siendo una URI.

Este elemento proporciona los medios para conseguir que un recurso de la web semántica sea identificado de manera unívoca.

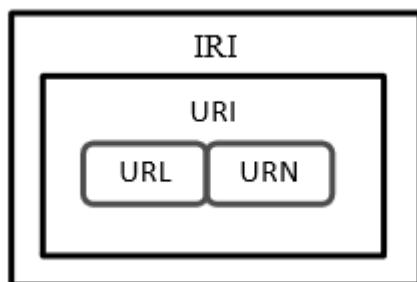


Ilustración 4: Estructura IRI

El mecanismo utiliza el URN como la clave para la identificación y la URL para proporcionar información sobre cómo y dónde encontrar los documentos en la web.

Una de las tareas difíciles es conseguir la estandarización de la multitud de conceptos que existen y asignar un identificador para cada objeto.

XML

Extensible markup language (**XML**) se define como un conjunto de reglas para la codificación de documentos. Es un formato de texto etiquetado que permite la creación de documentos compuestos de datos.

XML no permite definir relaciones entre las diferentes estructuras ni tampoco añade semántica.

Tecnologías de estandarización de la web semántica

RDF

RDF es un lenguaje para la definición de ontologías y metadatos en la web. Su primera versión fue publicada en 1999. Actualmente es el estándar con mayor aceptación en la web semántica. Este lenguaje tiene como objetivo crear modelos de información abiertos, para permitir a los agentes software de automaticen tareas de acceso y combinación de la información basándose en el significado que RDF da a los diferentes elementos.

La estructura de una expresión en RDF se forma gracias a tres elementos:

- Sujeto
- Predicado
- Objeto

Se puede representar de manera gráfica como se muestra en la siguiente figura:

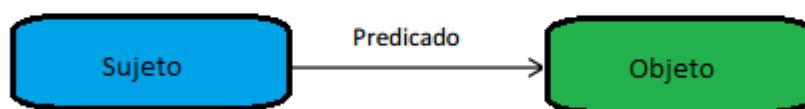


Ilustración 5: Expresión RDF

Un conjunto de tuplas como la que representa la figura anterior, forman un grafico RDF. Este grafico se puede asemejar a un mapa conceptual de un determinado dominio.

En RDF se usan tipos para definir los valores que puede tomar que puede tomar un nodo (persona, empresa, establecimiento...). Los tipos están compuestos por un espacio de valores en un espacio léxico y un mapeo (léxico-valor). Esto consigue la creación de sinónimos, facilitando y ampliando el uso de tecnología., cada elemento del espacio de valores puede estar relacionado de 0 a N elementos del espacio léxico y a su vez cada elemento del espacio léxico está relacionado con un elemento del espacio de valores.

Espacio de valores	{Persona, Empresa}
Espacio léxico	{Humano, business, establecimiento, negocio}
Mapeo I(léxico-valor)	{<persona, humano>; <empresa, business>; <empresa, establecimiento>; <empresa, negocio>}

Tabla 1: Tipos RDF

RDFS

Este lenguaje es una extensión de RDF (RDF Schema). Otorga unas propiedades parecidas a la programación orientada a objetos, permitiendo así dotar de mayor semántica a un contenido web. Aparece el concepto clase y propiedad y con ellos se pueden crear jerarquías y vocabularios.

Las clases se agrupan a un conjunto de recursos y a cada miembro se le llama instancia. Cada clase es un recurso que define sus propiedades mediante RDF, se identifica normalmente con una URI. El conjunto de instancias de una clase es la extensión de clase, pero una instancia, puede pertenecer a varias extensiones de clase (En la programación orientada a objetos ocurre lo contrario).

Una clase posee todas las instancias de la extensión de clase de sus subclases. Todas las clases descritas en RDFS son subclases de Resource la cual es una instancia de class.

Las propiedades son otro concepto importante que añade RDFS, las cuales también pueden tener subpropiedades. Por ejemplo, si un objeto R está relacionado con la propiedad P y esta propiedad es una subpropiedad P', R está relacionado con P'.

Los ámbitos y los dominios nos permiten formar un vocabulario integrado por jerarquías de clases y propiedades. La pertenencia a un ámbito implica que todos aquellos objetos expresados en RDFS con el mismo predicado pertenecen a la clase especificada.

SPARQL

Se trata de un lenguaje capaz de realizar consultas en diversas fuentes de datos almacenados como RDF, la información se presenta en un conjunto de resultados o gráficas.

Este lenguaje permite realizar consultas con temas de sujeto, predicado y objeto sin la necesidad de recibirlos todos informados, es decir puede dejar libre cualquiera de estas tres variables. Con esto se puede formular consultas que devuelvan todos los objetos que contienen por sujeto a "Cervantes" y como predicado "escrito por" lo que devolvería un conjunto de objetos "Libro".

La estructura de una consulta en SPARQL es:

- SELECT<var> WHERE { <sujeto><predicado><objeto>}

Una ventaja de SPARQL es que este incorpora herramientas para el filtrado de búsqueda de campos de texto, usando expresiones regulares y variables numéricas, mediante operadores aritméticos. Por lo que podremos consultar todos los objetos con predicado “Vendido en” y como predicado “Supermercado”, devolviendo como resultado de la búsqueda todos los objetos que han sido vendidos en supermercado.

Otra ventaja es que está basado en SQL, por lo que hay muchas sentencias que hereda de este, tales como, ORDER BY, DISTINCT, OFFSET o LIMIT, entre otras.

OWL

Antes de definir OWL (Ontology Web Language) es necesario definir el concepto de ontología.

Una ontología define los términos a utilizar para describir y representar un área de conocimiento. Las ontologías las utilizan las personas, las bases de datos e incluso las aplicaciones que tienen la necesidad de compartir un dominio de información (se conoce por dominio a un área de temática específica o un área de conocimiento, por ejemplo el área financiera o la medicina). Las ontologías incorporan definiciones de conceptos básicos del dominio, y las relaciones entre ellos, esto resulta bastante útil para los ordenadores.

Las ontologías codifican el conocimiento de un dominio y también el conocimiento que extiende de los dominios, por lo que podríamos concluir que hacen el conocimiento reutilizable.

OWL se considera como una extensión de RDF, ya que este se puede formular sobre RDF. OWL además de incluir la capacidad expresiva de RDFS, la extiende con la posibilidad de usar expresiones lógicas. Con el uso de OWL se puede, por ejemplo, definir clases mediante condiciones sobre sus miembros, con el uso de una combinación booleana de clases o por enumeración de las instancias que pertenecen a la clase. Este lenguaje también permite atribuir ciertas propiedades a las relaciones, como cardinalidad, simetría, transitividad, o relaciones inversas. OWL dota a los recursos de una semántica completa para que la información sea expuesta por RDFS.

Existen tres niveles de lenguaje OWL:

- OWL lite
- OWL DL
- OWL Full

Respectivamente ganan calidad descriptiva y especificativa en lo referido a la semántica, pero se hacen más complejos computacionalmente a medida que aumenta su nivel. Los lenguajes. Los lenguajes de menor nivel son un subconjunto de los de mayor nivel, es decir, OWL lite es un subconjunto de OWL DL y OWL DL es un subconjunto de OWL Full. A la hora de desarrollar un sistema hay que elegir entre un compromiso entre fidelidad del dominio u ontología y la carga computacional.

2.6 Análisis de metodologías para realizar el proyecto

2.6.1 Metodologías ágiles

Las Metodologías Ágiles forman un nuevo enfoque en el desarrollo de software, este nuevo enfoque suele tener mejor aceptación entre los desarrolladores de proyectos que las metodologías tradicionales (CMM,ISO-9000,etc.).Esto se debe a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeños tamaño, su ideología de colaboración Y su flexibilidad ante los cambios.

Las metodologías de desarrollo ágiles son adaptativas en lugar de predictivas. Los métodos ingeniería planean gran parte del proceso del software en detalle en un periodo de tiempo largo, esto funciona bien hasta que surgen cambios. Su naturaleza es resistirse al cambio. Para estas metodologías, no obstante, el cambio es bienvenido.

Las metodologías ágiles Intentan ser procesos que se adaptan y crecen en el cambio, hasta tal punto de cambiarse ellos mismos. Estas metodologías son orientadas a la gente y no orientadas al proceso. Su meta es definir un proceso que funcionará bien con cualquiera que lo use.

2.6.1.1 The Agile Alliance

En febrero de 2001, tras una reunión celebrada en EE.UU, se crea el término ágil aplicado al desarrollo de software. Participo Un grupo de expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue definir los valores y principios que deberían permitir a los equipos de trabajo desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. El objetivo era ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se fundó “The Agile Alliance”, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía ágil, es este manifiesto se valora:

- Desarrollar software que funciona más que conseguir una buena documentación. La regla seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.
- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte

automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta
- La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

Una de las características principales dentro de las metodologías ágiles es el uso de historias de usuario. Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

2.6.1.2 SCRUM

Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades.

Los principales beneficios que proporciona la utilización de Scrum son los siguientes:

- Cumplimiento de expectativas: El cliente establece sus expectativas indicando el valor que le aporta cada historia del proyecto, el equipo de trabajo estima y con esta información establece su prioridad. Un Sprint tiene demos, donde se comprueba que efectivamente los requisitos se han cumplido y se transmite feedback (retroalimentación) al equipo.
- Flexibilidad a cambios: Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado.
- Predicciones de tiempos: Se puede estimar con facilidad cuando estará disponible una determinada funcionalidad.
- El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- Mayor calidad del software: La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.

- Mayor productividad: Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
- Produce software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.

2.6.2 Metodología tradicional

Las Metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Entre las metodologías tradicionales se va a proceder a describir las siguientes:

2.6.2.1 RUP (*Rational Unified Procces*)

El Rational Unified Process (RUP) es un proceso de desarrollo de software elaborado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

Este proceso de desarrollo de software no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

El RUP está basado en los siguientes principios clave:

- **Adaptar el proceso:** el proceso debe adaptarse a las necesidades del cliente ya que es de vital importancia interactuar con él.
- **Equilibrar prioridades:** los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un equilibrio que satisfaga los deseos de todos.

- **Demostrar valor iterativamente:** los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.
- **Colaboración entre equipos:** El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requisitos, desarrollo, evaluaciones, planes y resultados.
- **Elevar el nivel de abstracción:** un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Éstas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.
- **Enfocarse en la calidad:** el control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Principales características.

- Implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software
- Asignar tareas y responsabilidades Forma disciplinada
- Administración de requisitos

Fases internas de RUP

RUP comprende dos aspectos importantes por los cuales se establecen las disciplinas:

1. **'Proceso'**: se divide en:

- Modelado de negocio
- Requisitos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue

2. **Soporte**: cuyas etapas son:

- Gestión del cambio y configuraciones
- Gestión del proyecto
- Entorno

La estructura dinámica de RUP permite que éste sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las cuatro fases descritas anteriormente:

1. **Fase de Inicio:** Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.
2. **Fase de Elaboración:** En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.
3. **Fase de Desarrollo:** El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
4. **Fase de Transición:** El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.



Ilustración 6: RUP estructura dinámica

2.6.2.2 Modelo en cascada.

Es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase.

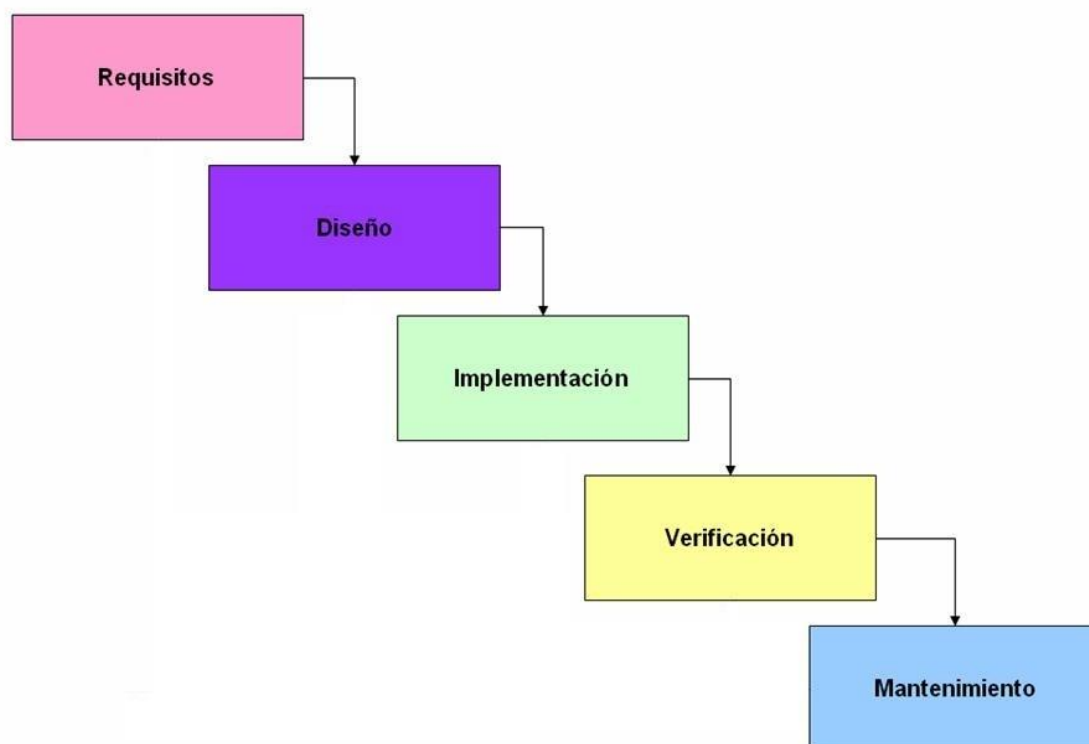


Ilustración 7: Fases del método en cascada

Análisis de requisitos

En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos.

Diseño del Sistema

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación.

Diseño del Programa

Fase en la cual se realizan los algoritmos necesarios para el cumplimiento tanto de los requerimientos del usuario, como también los análisis necesarios para saber las herramientas usadas en la etapa de Codificación

Codificación

Es la fase en la cual se implementa el código fuente, haciendo uso tanto de prototipos como de pruebas para corregir errores.

Pruebas

Los elementos programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que estos cumplen con los requisitos. Esto es antes de ser entregado al usuario final.

Verificación

Fase en la cual el usuario final ejecuta el sistema, previamente los programadores ya han realizado pruebas para comprobar que el sistema no tenga errores.

Mantenimiento

Es una de las etapas más críticas, ya que se destina una gran parte de los recursos, es el mantenimiento del Software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas.

Ventajas método en cascada

- Realiza un buen funcionamiento en equipos débiles y productos maduros, por lo que se requiere de menos capital y herramientas para hacerlo funcionar de manera óptima.
- Es un modelo fácil de implementar y entender.
- Está orientado a documentos.
- Es un modelo conocido y utilizado con frecuencia.
- Promueve una metodología de trabajo efectiva: Definir antes que diseñar, diseñar antes que codificar.

2.6.3 Comparativa entre metodologías ágiles y tradicionales

Para finalizar este apartado, se va a realizar una comparativa entre las metodologías analizadas, en la siguiente tabla se muestra los puntos fuertes de cada una:

Metodologías tradicionales o No ágiles	Metodologías Ágiles
Contrato prefijado.	No existe un contrato prefijado, este suele ser bastante flexible.
El cliente interactúa por reuniones con el equipo de trabajo.	El cliente es parte del equipo de desarrollo.
Énfasis en la arquitectura.	Menos énfasis en la arquitectura
Grupos de trabajo grandes y trabajando en diferentes lugares.	Grupos de trabajos pequeños (Menos de 10 trabajadores), trabajando en el mismo lugar.
Más roles.	Menos roles
Muchos artefactos.	Pocos Artefactos

Tabla 2: Comparativa metodologías ágiles y tradicionales

2.7 Solución propuesta

En este apartado se describen las tecnologías, herramientas y metodología escogidas para la realización de este proyecto.

2.7.1 Herramientas y tecnologías

BOOTSTRAP

Para la parte de presentación vamos hacer uso del framework bootstrap.



Ilustración 8: Bootstrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

El beneficio de usar responsive design en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda. El uso de *media queries*, que es un módulo de CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla y sí se trabaja las dimensiones de tu contenido en porcentajes, puedes tener una web muy fluida capaz de adaptarse a casi cualquier tamaño de forma automática.

Bootstrap utiliza los siguientes lenguajes de programación:

- HTML5
- CSS3
- JavaScript

A continuación se exponen algunas de las ventajas propias de bootstrap:

- Mobile First. Todo el framework está desarrollado pensando en la filosofía mobilefirst, que es la recomendada para el desarrollo de sitios web responsive.
- Descarga personalizable. No es necesario bajarse todo el framework para que funcione, ni siquiera requiere JavaScript para las funcionalidades básicas, podemos personalizar la descarga y dejar de lado lo que no necesitamos.
- Optimizado para LESS. Si trabajas con LESS, un potente preprocesador de CSS, entonces el mejor framework que tienes a tu disposición es sin duda alguna Bootstrap.
- Excelente grid responsive. El grid de Bootstrap es 100% responsive y te permite pensar en el diseño en todas sus vistas, de modo que ofrezcas la mejor experiencia de usuario a todo el mundo.
- Compatibilidad entre navegadores. Está desarrollado para funcionar con todos los navegadores web modernos, dando soporte incluso a Internet Explorer 8 y superiores.
- Uso de Font icons. Con Bootstrap se dejan de usar los sprites de CSS para las imágenes y se pasa a usar los Font icons, que son vectoriales, escalables y óptimos para rendimiento web.
- Excelente documentación. La documentación online de Bootstrap es de muy alta calidad, con ejemplos muy claros y simples y código reutilizable.
- Comunidad online. Tal y como comentaba más arriba, Bootstrap cuenta con una gran comunidad online, siendo tendencia en el repertorio de Github.

MongoDB

Para la persistencia de datos usaremos MongoDB



Ilustración 9: MongoDB

MongoDB es una base de datos NoSQL orientada a documentos, desarrollada con código abierto.

NoSQL, Significa que en lugar de guardar los datos en tablas como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema dinámico (BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

En la imagen inferior podemos ver como MongoDB se encuentra en la zona óptima, en la que la velocidad y la escalabilidad son altas, así como también es el número de objetos de la

base de datos. Por contra en una base de datos relacional (RDBMS) tanto la escalabilidad como la velocidad se ven penalizados ante un número elevado de elementos en la base de datos.

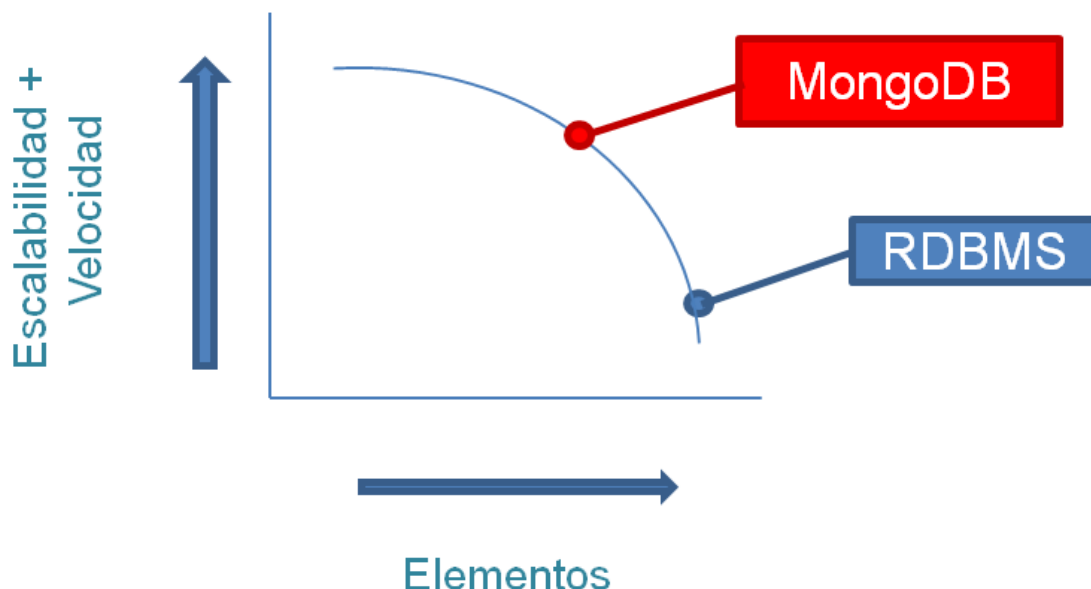


Ilustración 10: Escalabilidad + Velocidad MongoDB

MongoDB está escrito en C++, aunque las consultas se hacen pasando objetos JSON como parámetro. Es algo bastante lógico, dado que los propios documentos se almacenan en BSON. Por ejemplo:

```
db.Clientes.find({Nombre:"Pedro"});
```

La consulta anterior buscará todos los clientes cuyo nombre sea Pedro.

MongoDB viene de serie con una consola desde la que podemos ejecutar los distintos comandos. Esta consola está construida sobre JavaScript, por lo que las consultas se realizan utilizando ese lenguaje. Además de las funciones de MongoDB, podemos utilizar muchas de las funciones propias de JavaScript. En la consola también podemos definir variables, funciones o utilizar bucles.

Herramienta de desarrollo



Ilustración 11: Spring

Spring es un framework ligero para construir aplicaciones empresariales. Spring se encuentra dividido en distintos módulos, cada uno de los cuales se encarga de partes diferentes de nuestra aplicación, en aplicaciones grandes o realmente grandes puede ahorrarnos mucho trabajo ya que puede coordinar todas las partes de la aplicación. Esta separación en módulos nos permite usar solo las partes que necesitamos, sin tener la carga de los que no usamos.

Spring está diseñado para no ser intrusivo, esto significa que no es necesario que nuestra aplicación extienda o implemente alguna clase o interface de Spring (si no lo queremos), por lo que nuestro código de lógica quedará libre y completamente reutilizable para un proyecto sin Spring, o por si debemos quitarlo de una aplicación que ya lo esté usando. Gracias a esto es posible usar un POJO o un objeto Java para hacer cosas que antes solo podían hacerse con EJBs. Sin embargo la utilidad de Spring no es solo para el desarrollo de aplicaciones web, o no solo en el servidor. Cualquier aplicación Java puede beneficiarse del uso de Spring.

Al usar Spring de la forma correcta nuestra aplicación quedará dividida en capas bien delimitadas, y con buenas prácticas de programación.

El núcleo de Spring está basado en un principio o patrón de diseño llamado Inversión de Control (IoC por sus siglas en inglés). Las aplicaciones que usan el principio de IoC se basan en su configuración (que en este caso puede ser en archivos XML o con anotaciones como en Hibernate) para describir las dependencias entre sus componentes, esto es, los otros objetos con los que interactúa. En este caso “inversión” significa que la aplicación no controla su estructura; permite que sea el framework de IoC (en este caso Spring) quien lo haga.

El uso de interfaces y DI son mutuamente benéficos, ya que hace más flexible y robusta nuestra aplicación y es mucho más fácil realizar pruebas unitarias. Pero la complejidad de escribir código que maneje las dependencias entre los componentes de una aplicación diseñada para usar interfaces puede llegar a ser mucho y esto, además, hace que los desarrolladores tengamos que escribir aún más código.

Afortunadamente, usando DI reducimos la cantidad de código extra que debemos escribir, para un diseño basado en interfaces, casi a cero.

En el contexto de DI, Spring actúa como un contenedor que proporciona las instancias de las clases de nuestra aplicación todas las dependencias que necesita, pero en una forma no intrusiva y automática. Todo lo que debemos hacer es crear un archivo de configuración que describa las dependencias; Spring se hará cargo del resto.

Spring es un contenedor ya que no solo crea los componentes de nuestra aplicación, sino porque contiene y maneja al ciclo de vida y configuración de estos componentes. En Spring, podemos declarar cómo debe ser creado cada uno de los objetos de nuestra aplicación, cómo deben ser configurados, y cómo deben asociarse con los demás.

La implementación de DI de Spring se enfoca en el acoplamiento débil: los componentes de nuestra aplicación deben asumir lo menos posible acerca de otros componentes. La forma más fácil de lograr este bajo acoplamiento en Java es mediante el uso de Interfaces. Como cada componente de la aplicación solo está consciente de la interface de otros componentes, podemos cambiar la implementación del componente sin afectar a los componentes que usan el nuevo componente. Hablaré un poco más de esto cuando veamos los ejemplos.

El uso de DI tiene como beneficios, además de lo que ya he mencionado arriba, los siguientes:

- Reduce el código pegamento: Esto quiere decir que reduce dramáticamente la cantidad de código que debemos escribir para unir los distintos componentes. Aunque algunas veces este código puede ser tan simple como usar el operador “new” para instanciar un nuevo objeto, otras puede ser más complejo, como realizar una búsqueda de dicha dependencia en un repositorio a través de JNDI, como en el caso de los recursos remotos. En este caso, el uso de DI puede reducir de forma dramática la cantidad de código pegamento (o gluecode) proporcionando búsquedas automáticas.
- Externaliza dependencias: Como es posible colocar la configuración de dependencias en archivos XML podemos realizar una reconfiguración fácilmente, sin necesidad de recompilar nuestro código. Gracias a esto es posible realizar el cambio de la implementación de una dependencia a otra (como en el ejemplo de Hibernate que mencioné antes)
- Las dependencias se manejan en un solo lugar: Toda la información de dependencias es responsabilidad de un solo componente, el contenedor de IoC de Spring, haciendo que este manejo de dependencias más simple y menos propenso a errores.
- Hace que las pruebas sean más fáciles: Nuevamente, como nuestras clases serán diseñadas para que sea fácil el reemplazo de dependencias, podemos proporcionar mocks o dummies, que regresen datos de prueba, de servicios o cualquier dependencia que necesite el componente que estamos probando.

Como podemos ver, el uso de DI nos proporciona muchos beneficios, pero no sin sus correspondientes desventajas. En particular, es difícil ver qué implementación particular de una dependencia está siendo usada para qué objeto, especialmente para alguien que no está familiarizado con esta forma de trabajo.

Spring está dividido en alrededor de 20 módulos y colocados en los siguientes grupos:

- Contenedor Central (Core Container)
- Acceso a Datos / Integración
- WEB
- AOP (Programación Orientada a Aspectos)
- Instrumentación
- Pruebas

Estos grupos se muestran en la siguiente imagen:

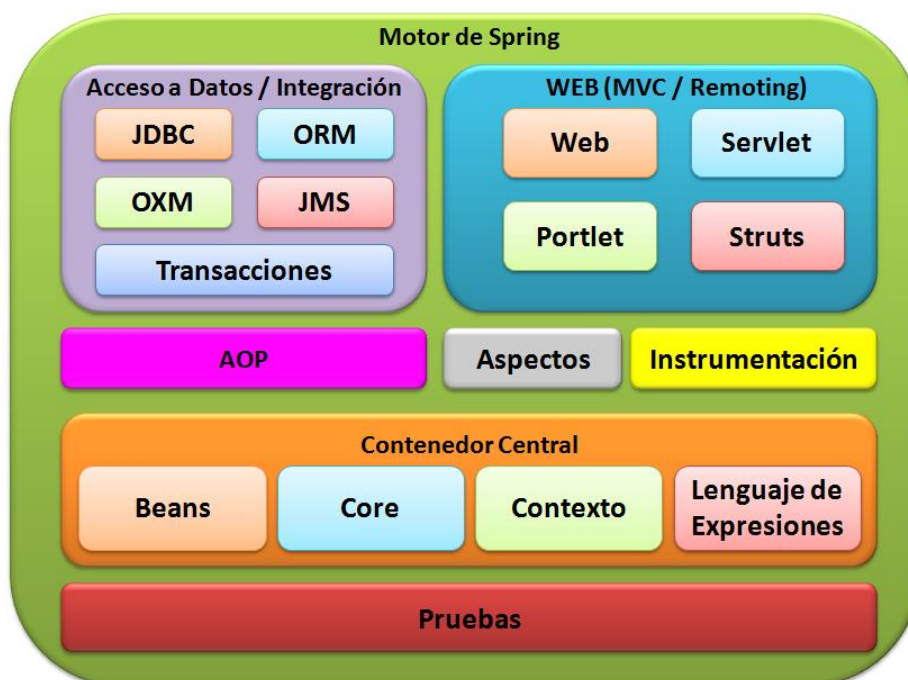


Ilustración 12: Motor Spring

A lo largo de estos tutoriales trataré de explicar la mayor cantidad de módulos. Pero como dije antes: Spring es en realidad un monstruo. Así que probablemente cuando termine de explicarlos todos, ya habrá salido una nueva versión, así que me centraré solo en los más importantes y en su integración con otros frameworks que comencemos a usar en otros tutoriales.

En general, estas son algunas de las características de Spring:

- Simplificación de la programación orientada a aspectos.
- Simplificación del acceso a datos.
- Simplificación e integración con JEE
- Soporte para planificación de trabajos.
- Soporte para envío de mail.
- Interacción con lenguajes dinámicos (como BeanShell, JRuby, y Groovy).
- Soporte para acceso a componentes remotos.
- Manejo de Transacciones.
- Su propio framework MVC.
- Su propio Web Flow.
- Manejo simplificado de excepciones.

Apache Marmotta



Ilustración 13: Apache Marmotta

Apache Marmotta es un servidor diseñado para ofrecer múltiples tecnologías de Apache en una sola instalación.

El proyecto tiene como objetivo implementar la aplicación Media Framework Vinculado a las organizaciones que necesitan publicar datos enlazados. El programa le proporciona un fácil utilizar el método de almacenamiento en caché Linked Data y permite acceder o modificar Linked Data para los proyectos.

2.7.2 Repositorio del proyecto



Ilustración 14: GitHub

Para poder trabajar desde varias máquinas y con el objetivo de mantener un código legible y actualizado, vamos hacer uso de un repositorio que se encargue de almacenar los cambios que se vayan generando en el código. Además queremos que nos ofrezca la posibilidad de ver los cambios realizados desde los diferentes puestos de trabajo conectados a nuestro repositorio.

Para llevar a cabo estas tareas se ha escogido la herramienta github, que hace uso del sistema de control de versiones git. Git es un sistema descentralizado, no necesita de un servidor donde subir los cambios, sino que los desarrolladores se envían los cambios unos a otros desde sus equipos.

GitHub es más que un servicio de alojamiento de código. Hoy en día ofrece varias herramientas en línea muy útiles para el trabajo en equipo. Entre ellas cabe destacar:

- Un wiki que funciona con gollum, el cual opera con Git para el mantenimiento de las distintas versiones de las páginas.
- Un sistema de seguimiento de problemas, que al estilo del clásico sistema de tickets, permiten a los miembros de tu equipo (o a cualquier usuario de GitHub si tu repositorio es público) abrir un ticket detallando un problema que tenga con tu software o una sugerencia que desee hacer al mismo.
- Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero (ej: "Esto es mejor que lo extraigamos a una nueva clase"), y debatir sobre determinados cambios realizados en un commit específico.
- Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

2.7.3 Metodología de desarrollo

Hoy en día, cada vez es más habitual que las empresas dedicadas al desarrollo de software, hagan uso de una metodología ágil a la hora de realizar sus procesos de desarrollo.

Estas metodologías no solo han conseguido un cambio en la manera de desarrollar, sino también un cambio organizativo en las empresas en las que se está implantando.

A pesar de las ventajas de hacer uso de las metodologías ágiles, muchas empresas en la actualidad siguen haciendo uso de las metodologías tradicionales. Las cuales tienen problemas en el desarrollo, cuando existen cambios requeridos por el cliente, ya que este habitualmente tienden a saber qué resultados quiere para su producto final una vez que lo han visto.

Con la realización de este TFG se pretende experimentar como afecta el uso de una determinada metodología a la hora de realizar el proceso de desarrollo de nuestro software. Para poder tener una visión objetiva de las ventajas de las tecnologías ágiles, antes se debe experimentar con los problemas o beneficios del uso de las metodologías tradicionales.

En este TFG se va a usar una metodología de desarrollo en cascada, pero con una variante. Se van a dejar los requisitos de usuario abiertos, por lo se podrán ir modificando a lo largo del desarrollo del proyecto. Con el uso de esta metodología se pretende experimentar las dificultades que se pueden presentar en el desarrollo de un producto haciendo uso de esta metodología.

3 ANÁLISIS

3.1 Descripción general

El objetivo de este proyecto es implementar un sistema de gestión de grupos de investigación que haga uso de las tecnologías semánticas.

En este apartado se van a definir los requisitos de usuario, casos de uso y requisitos software de nuestro sistema a implementar.

Con esto se conseguirá delimitar las funciones que debe implementar nuestro sistema a desarrollar.

3.2 Requisitos de usuario

A continuación siguiendo con el análisis se van a definir los requisitos de usuario. Estos deben ser descritos por el cliente, en este caso en particular será el alumno el cual desempeñara esta función.

Para ello se va a utilizar el siguiente patrón.

Identificador ¹			
Título ²			
Descripción ³			
Prioridad ⁴		Estabilidad ⁵	
Claridad ⁶		Verificabilidad ⁷	
Necesidad ⁸		Fuente ⁹	

Tabla 3: Patrón requisito de usuario

Descripción de la tabla:

1. **Identificador:** Código que identifica de forma unívoca cada requisito. Seguirá el siguiente formato: RUX-YY donde X es la inicial del tipo del requisito Capacidad o Restricción e YY es un número correlativo incremental.
2. **Título:** Título descriptivo del requisito.
3. **Descripción:** Explicación clara y concisa del requisito que se está especificando.

4. **Prioridad:** Orden de cumplimiento de un requisito. Puede tomar los siguientes valores: baja, media o alta.
5. **Estabilidad:** Probabilidad de cambio de un requisito. Puede tomar los valores: estable o inestable.
6. **Claridad:** Campo para representar la no ambigüedad de un requisito. Puede tomar los valores: baja, media o alta.
7. **Verificabilidad:** Facilidad para comprobar que el sistema cumple un requisito. Puede tomar los valores: baja, media o alta.
8. **Necesidad:** Interés del cliente en la realización de un requisito. Puede tomar los siguientes valores: baja, media o alta.
9. **Fuente:** Origen del requisito.

3.3.1 Requisitos de capacidad

RUC-01			
Titulo	Acceso a la aplicación web		
Descripción	El usuario puede acceder a la aplicación web usando sus credenciales de acceso.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 4: RUC-01, Acceso a la aplicación web

RUC-02			
Titulo	Visualizar Proyectos Asignados		
Descripción	El usuario conectado puede visualizar los proyectos en los cuales está asignado.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 5: RUC-02 , Visualizar Proyectos Asignados

RUC-03			
Titulo	Visualizar un proyecto		
Descripción	El usuario podrá visualizar un proyecto permitiéndole: <ul style="list-style-type: none"> Visualizar información del proyecto Gestionar gastos Gestionar conferencias 		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 6: RUC-03, Visualizar un proyecto

RUC-04			
Titulo	Visualizar información del proyecto		
Descripción	El usuario podrá ver la información relacionada con el proyecto.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 7: RUC-04, Visualizar información del proyecto

RUC-05			
Titulo	Gestionar gastos		
Descripción	El usuario podrá gestionar los gastos de un proyecto: <ul style="list-style-type: none"> Consultar gastos Añadir Gastos 		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 8: RUC-05, Gestionar gastos

RUC-06			
Titulo	Consultar gastos		
Descripción	El usuario podrá consultar los gastos del proyecto seleccionado.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 9: RUC-06, Consultar gastos

RUC-07			
Titulo	Añadir Gastos		
Descripción	El usuario podrá asignar gastos al proyecto seleccionado.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 10: RUC-07, Añadir Gastos

RUC-08			
Titulo	Gestionar conferencias		
Descripción	El usuario podrá gestionar las conferencias de un proyecto: <ul style="list-style-type: none"> • Consultar conferencias • Añadir conferencias (solo jefe del proyecto) 		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 11: RUC-08, Gestionar conferencias

RUC-09			
Titulo	Consultar conferencias		
Descripción	El usuario podrá visualizar las conferencias del proyecto seleccionado.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 12: RUC-09, Consultar conferencias

RUC-10			
Titulo	Añadir conferencias		
Descripción	El usuario podrá añadir conferencias al proyecto seleccionado. Para poder añadir una conferencia el usuario debe ser el jefe del proyecto.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 13: RUC-10, Añadir conferencias

RUC-11			
Titulo	Buscar un proyecto		
Descripción	El usuario podrá buscar proyectos. Este buscador deberá hacer uso de las tecnologías semánticas.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 14: RUC-11, Buscar un proyecto

3.3.2 Requisitos de restricción

RUR-01			
Titulo	Idioma		
Descripción	El idioma de la aplicación debe ser el castellano.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 15: RUR-01, Idioma

RUR-02			
Titulo	Disponibilidad en varios navegadores		
Descripción	El sistema debe estar disponible en los principales navegadores web.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 16: RUR-02, Disponibilidad en varios navegadores

RUR-03			
Titulo	Disponibilidad horaria		
Descripción	La página debe estar disponible las 24 horas del día.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 17: RUR-03, Disponibilidad horaria

RUR-04			
Titulo	Interfaz sencilla		
Descripción	La interfaz debe facilitar la navegación del usuario en la aplicación.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	Cliente

Tabla 18: RUR-04, Interfaz sencilla

3.3 Casos de uso

Los casos de uso detallan que acciones realiza un sistema desde el punto de vista del usuario, describiendo el uso del sistema y como este interactúa con los usuarios. Con ello conseguimos describir de una forma más clara lo que hace el sistema.

A continuación se muestra una descripción visual y contextual de los casos de uso.

3.3.1 Diagramas UML

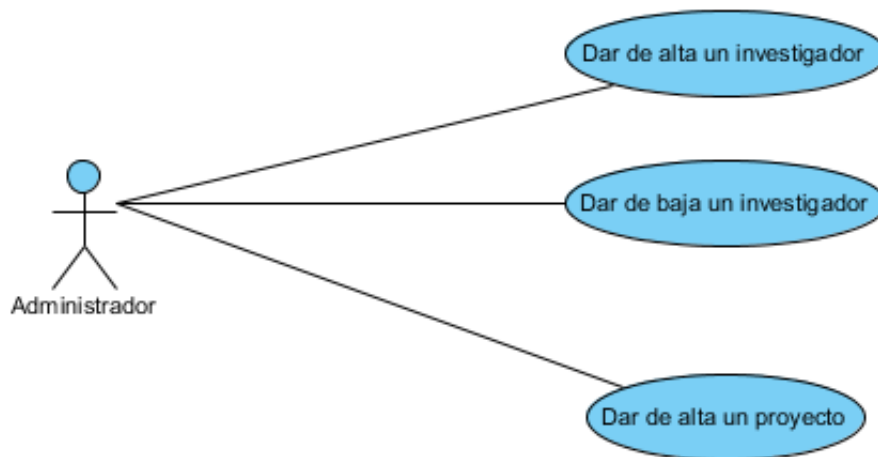


Ilustración 15: Diagrama administrador

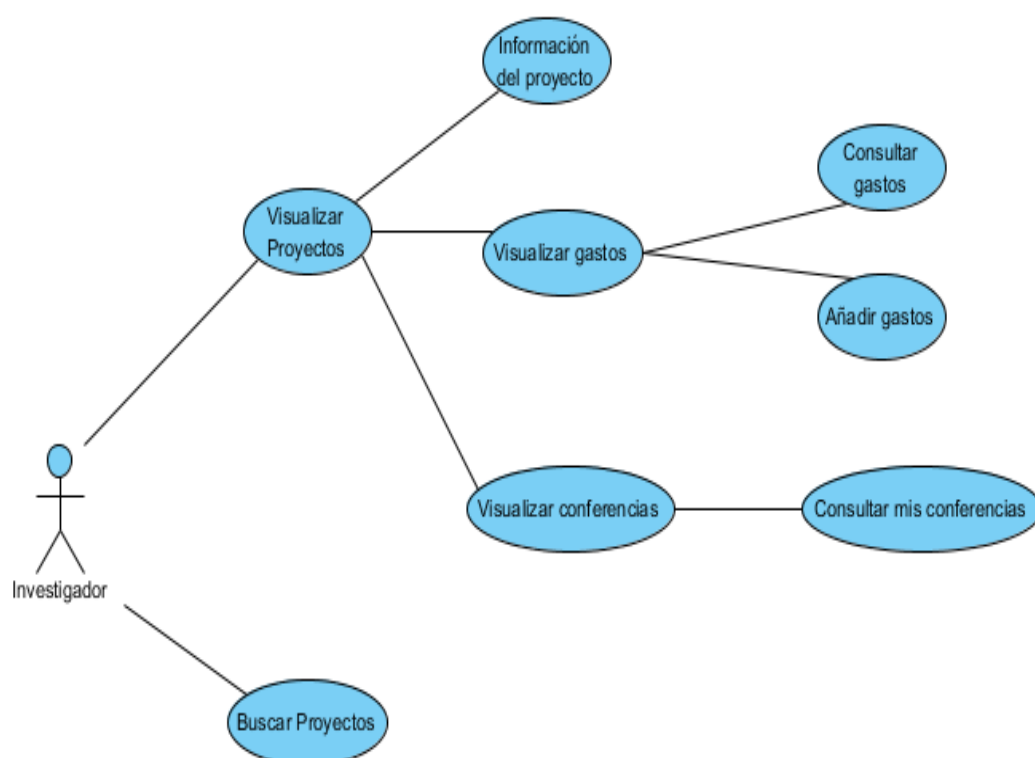


Ilustración 16: Diagrama Investigador

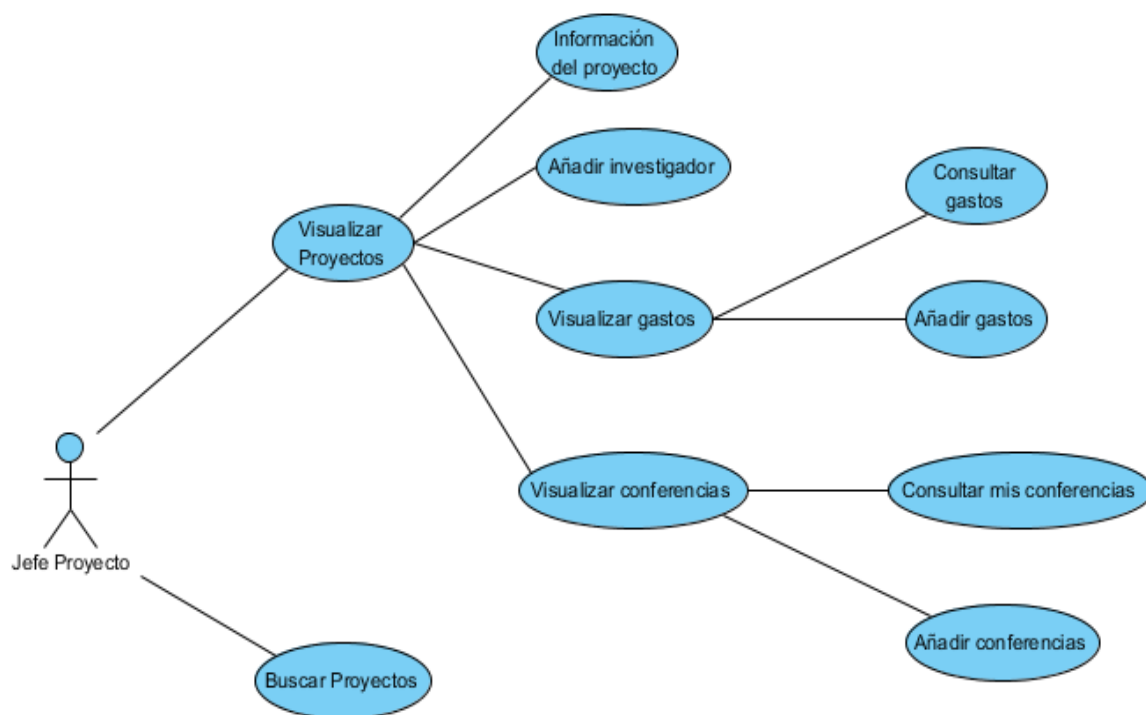


Ilustración 17: Diagrama Jefe de proyecto

3.3.2 Descripción contextual

Para realizar dicha descripción vamos hacer uso de una plantilla para facilitar su legibilidad:

Identificador ¹	
Nombre ²	
Actor ³	
Descripción ⁴	
Flujo ⁵	
Precondiciones ⁶	
Post-condiciones ⁷	

Tabla 19: Plantilla casos de uso

1. **Identificador:** Código que identifica unívocamente a cada caso de uso. Seguirá el siguiente formato: CUX-YY donde X es la inicial del Actor (Investigador, jefe de proyecto o administrador) e YY es un número correlativo incremental.
2. **Nombre:** Nombre descriptivo del caso de uso.
3. **Descripción:** Explicación clara del caso de uso.
4. **Actor:** Tipo de usuario que usa la aplicación.
5. **Precondiciones:** Condiciones previas que deben ser cumplidas para poder realizar una determinada operación.
6. **Post-condiciones:** Estado que presenta el sistema tras la ejecución de una determinada operación.

3.3.2.1. Casos de uso: Usuario administrador

A continuación se describen los casos de uso del usuario administrador (CUA-XX).

CUA-01	
Nombre	Dar de alta un investigador
Actor	Usuario administrador
Descripción	El usuario administrador puede dar de alta un nuevo investigador en la aplicación.
Flujo	<ol style="list-style-type: none">1. El usuario accede a la web.2. Introduce sus credenciales para logarse como usuario administrador.3. En el menú selecciona la opción "Dar de alta un investigador".4. Rellena el formulario y pulsa el botón "enviar".
Precondiciones	El usuario administrador debe tener el sistema implantado.
Post-condiciones	No aplica.

Tabla 20: CUA-01, Dar de alta un investigador

CUA-02	
Nombre	Dar de baja un investigador
Actor	Usuario administrador
Descripción	El usuario administrador puede dar de baja a un investigador en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario administrador. 3. En el menú selecciona la opción "Dar de baja un investigador". 4. Rellena el formulario y pulsa el botón "enviar".
Precondiciones	El usuario administrador debe tener el sistema implantado.
Post-condiciones	No aplica.

Tabla 21: CUA-02, Dar de baja un investigador

CUA-03	
Nombre	Dar de alta un Proyecto
Actor	Usuario administrador
Descripción	El usuario administrador puede dar de alta un nuevo proyecto en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario administrador. 3. En el menú selecciona la opción "Dar de alta un Proyecto". 4. Rellena el formulario y pulsa el botón "enviar".
Precondiciones	El usuario administrador debe tener el sistema implantado.
Post-condiciones	No aplica.

Tabla 22: CUA-03, Dar de alta un Proyecto

3.3.2.2. Casos de uso: Usuario investigador

En este apartado se describen los casos de uso del usuario investigador (CUI-XX).

CUI-01	
Nombre	Visualizar proyectos.
Actor	Usuario investigador.
Descripción	El usuario investigador puede visualizar proyectos en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario investigador. 3. En el menú selecciona la opción "Visualizar proyectos".
Precondiciones	El usuario investigador debe estar dado de alta en el sistema.
Post-condiciones	CUI-02, CUI-03, CUI-06

Tabla 23: CUI-01, Visualizar proyectos.

CUI-02	
Nombre	Ver información del proyecto.
Actor	Usuario investigador.
Descripción	El usuario investigador puede visualizar la información de un proyecto en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario investigador. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú selecciona la opción "Ver información del proyecto".
Precondiciones	CUI-01
Post-condiciones	No aplica.

Tabla 24: CUI-02, Ver información del proyecto.

CUI-03	
Nombre	Visualizar gastos.
Actor	Usuario investigador.
Descripción	El usuario investigador puede visualizar los gastos.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario investigador. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú selecciona la opción "Visualizar gastos".
Precondiciones	CUI-01
Post-condiciones	CUI-04. CUI-05

Tabla 25: CUI-03, Visualizar gastos.

CUI-04	
Nombre	Consultar gastos.
Actor	Usuario investigador.
Descripción	El usuario investigador puede consultar los gastos de un proyecto.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario investigador. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú selecciona la opción "Visualizar gastos". 5. En el menú elige la opción "Consultar gastos".
Precondiciones	CUI-03
Post-condiciones	No Aplica.

Tabla 26: CUI-04, Consultar gastos.

CUI-05	
Nombre	Añadir gastos.
Actor	Usuario investigador.
Descripción	El usuario investigador puede añadir gastos a un proyecto.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario investigador. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú selecciona la opción "Visualizar gastos". 5. En el menú elige la opción "Añadir gastos". 6. Rellena el formulario y pulsa aceptar
Precondiciones	CUI-03
Post-condiciones	No Aplica.

Tabla 27: CUI-05, Añadir gastos.

CUI-06	
Nombre	Visualizar conferencias.
Actor	Usuario investigador.
Descripción	El usuario investigador puede visualizar las conferencias asociadas a un proyecto.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario investigador. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú seleccionar la opción "Visualizar conferencias".
Precondiciones	CUI-01
Post-condiciones	CUI-07

Tabla 28: CUI-06, Visualizar conferencias.

CUI-07	
Nombre	Consultar mis conferencias.
Actor	Usuario investigador.
Descripción	El usuario investigador puede visualizar las conferencias que tiene asociadas.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario investigador. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú seleccionar la opción "Visualizar conferencias". 5. En el menú, seleccionar la opción "Consultar mis conferencias".
Precondiciones	CUI-07
Post-condiciones	No Aplica.

Tabla 29: CUI-07, Consultar mis conferencias.

CUI-08	
Nombre	Buscar proyectos.
Actor	Usuario investigador.
Descripción	El usuario investigador puede buscar un proyecto en la aplicación
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario investigador. 3. En el menú selecciona la opción "buscar proyectos". 4. A continuación en el buscador introducir el nombre del proyecto a buscar.
Precondiciones	El usuario investigador debe estar dado de alta en el sistema.
Post-condiciones	No Aplica.

Tabla 30: CUI-08, Buscar proyectos.

3.3.2.3. Casos de uso: Usuario Jefe Proyecto

En este apartado se describen los casos de uso del usuario Jefe Proyecto (CUJ-XX).

CUJ-01	
Nombre	Visualizar proyectos.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto puede visualizar proyectos en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción “Visualizar proyectos”.
Precondiciones	El usuario investigador debe estar dado de alta en el sistema.
Post-condiciones	CUJ-02, CUJ-03, CUJ-04, CUJ-07

Tabla 31: CUJ-01, Visualizar proyectos.

CUJ-02	
Nombre	Ver información del proyecto.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto puede visualizar la información de un proyecto en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción “Visualizar proyectos”. 4. A continuación en el menú selecciona la opción “Ver información del proyecto”.
Precondiciones	CUJ-01
Post-condiciones	No aplica.

Tabla 32: CUJ-02, Ver información del proyecto.

CUJ-03	
Nombre	Añadir investigador.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto añadir un investigador al proyecto.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú selecciona la opción "Añadir investigador". 5. Rellena el formulario y pulsa "Enviar".
Precondiciones	CUJ-01
Post-condiciones	No aplica.

Tabla 33: CUJ-03, Añadir investigador.

CUJ-04	
Nombre	Visualizar gastos.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto puede visualizar los gastos.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú selecciona la opción "Visualizar gastos".
Precondiciones	CUJ-01
Post-condiciones	CUJ-05. CUJ-06

Tabla 34: CUJ-04, Visualizar gastos.

CUJ-05	
Nombre	Consultar gastos.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto puede consultar los gastos de un proyecto.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú selecciona la opción "Visualizar gastos". 5. En el menú elige la opción "Consultar gastos".
Precondiciones	CUJ-04
Post-condiciones	No Aplica.

Tabla 35: CUJ-05, Consultar gastos.

CUJ-06	
Nombre	Añadir gastos.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto puede añadir gastos a un proyecto.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú selecciona la opción "Visualizar gastos". 5. En el menú elige la opción "Añadir gastos". 6. Rellena el formulario y pulsa aceptar
Precondiciones	CUJ-04
Post-condiciones	No Aplica.

Tabla 36: CUJ-06, Añadir gastos.

CUJ-07	
Nombre	Visualizar conferencias.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto puede visualizar las conferencias asociadas a un proyecto.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú seleccionar la opción "Visualizar conferencias".
Precondiciones	CUJ-01
Post-condiciones	CUJ-08, CUJ-09

Tabla 37: CUJ-07, Visualizar conferencias.

CUJ-08	
Nombre	Consultar mis conferencias.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto puede visualizar las conferencias que tiene asociadas.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú seleccionar la opción "Visualizar conferencias". 5. En el menú, seleccionar la opción "Consultar mis conferencias".
Precondiciones	CUJ-07
Post-condiciones	No Aplica.

Tabla 38: CUJ-08, Consultar mis conferencias.

CUJ-09	
Nombre	Añadir conferencias.
Actor	Usuario investigador.
Descripción	El usuario jefe proyecto puede visualizar las conferencias que tiene asociadas.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. Introduce sus credenciales para logarse como usuario jefe proyecto. 3. En el menú selecciona la opción "Visualizar proyectos". 4. A continuación en el menú seleccionar la opción "Visualizar conferencias". 5. En el menú, seleccionar la opción "Añadir conferencias". 6. Rellena el formulario y pulsa "enviar".
Precondiciones	CUJ-07
Post-condiciones	No Aplica.

Tabla 39: CUJ-09, Añadir conferencias.

CUI-10	
Nombre	Buscar proyectos.
Actor	Usuario jefe proyecto.
Descripción	El usuario jefe proyecto puede buscar un proyecto en la aplicación
Flujo	<ol style="list-style-type: none"> 5. El usuario accede a la web. 6. Introduce sus credenciales para logarse como usuario jefe proyecto. 7. En el menú selecciona la opción "buscar proyectos". 8. A continuación en el buscador introducir el nombre del proyecto a buscar.
Precondiciones	No Aplica.
Post-condiciones	No Aplica.

Tabla 40: CUJ-10, Buscar proyectos.

3.4 Requisitos de software

Después de definir los requisitos de usuario y los casos de uso, vamos a proceder a definir los requisitos software. En estos requisitos se definen las bases de la aplicación.

Para facilitar la lectura de los requisitos software usaremos la siguiente plantilla:

Identificador ¹			
Título ²			
Descripción ³			
Prioridad ⁴		Estabilidad ⁵	
Claridad ⁶		Verificabilidad ⁷	
Necesidad ⁸		Fuente ⁹	

Tabla 41: Plantilla Requisitos Software

Descripción de la tabla:

1. **Identificador:** Código que identifica de forma unívoca cada requisito. Seguirá el siguiente formato: RSF-XX o RSNF-XX, donde XX es un número correlativo incremental.
2. **Título:** Título descriptivo del requisito.
3. **Descripción:** Explicación clara y concisa del requisito que se está especificando.
4. **Prioridad:** Orden de cumplimiento de un requisito. Puede tomar los siguientes valores: baja, media o alta.
5. **Estabilidad:** Probabilidad de cambio de un requisito. Puede tomar los valores: estable o inestable.
6. **Claridad:** Campo para representar la no ambigüedad de un requisito. Puede tomar los valores: baja, media o alta.
7. **Verificabilidad:** Facilidad para comprobar que el sistema cumple un requisito. Puede tomar los valores: baja, media o alta.
8. **Necesidad:** Interés del cliente en la realización de un requisito. Puede tomar los siguientes valores: baja, media o alta.
9. **Fuente:** Origen del requisito.

3.4.1 Requisitos de software funcionales

RSF-01			
Titulo	Acceso a la aplicación web usuario administrador.		
Descripción	El sistema debe permitir acceder a la aplicación web a un usuario con credenciales de acceso de usuario administrador.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-01

Tabla 42: RSF-01, Acceso a la aplicación web usuario administrador.

RSF-02			
Titulo	Acceso a la aplicación web usuario investigador.		
Descripción	El sistema debe permitir acceder a la aplicación web a un usuario con credenciales de acceso de usuario investigador.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-01

Tabla 43: RSF-02, Acceso a la aplicación web usuario investigador.

RSF-03			
Titulo	Acceso a la aplicación web usuario jefe proyecto.		
Descripción	El sistema debe permitir acceder a la aplicación web a un usuario con credenciales de acceso de usuario jefe proyecto.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-01

Tabla 44: RSF-03, Acceso a la aplicación web usuario jefe proyecto.

RSF-04			
Titulo	Visualizar Proyectos Asignados		
Descripción	El sistema debe permitir visualizar los proyectos en los cuales un usuario está asignado.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-02

Tabla 45: RSF-04, Visualizar Proyectos Asignados

RSF-05			
Titulo	Visualizar un proyecto		
Descripción	El sistema debe permitir visualizar un proyecto a un usuario permitiéndole: <ul style="list-style-type: none"> • Visualizar información del proyecto • Gestionar gastos • Gestionar conferencias 		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-03

Tabla 46: RSF-05, Visualizar un proyecto

RSF-06			
Titulo	Visualizar información del proyecto		
Descripción	El sistema debe permitir al usuario visualizar ver la información relacionada con el proyecto.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-04

Tabla 47: RSF-06, Visualizar información del proyecto

RSF-07			
Titulo	Gestionar gastos		
Descripción	El sistema debe permitir al usuario gestionar los gastos de un proyecto: <ul style="list-style-type: none"> • Consultar gastos • Añadir Gastos 		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-05

Tabla 48: RSF-07, Gestionar gastos

RSF-08			
Titulo	Consultar gastos		
Descripción	El sistema debe permitir al usuario consultar los gastos del proyecto seleccionado.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-06

Tabla 49: RSF-08, Consultar gastos

RSF-10			
Titulo	Añadir Gastos		
Descripción	El sistema debe permitir al usuario asignar gastos al proyecto seleccionado.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-07

Tabla 50: RSF-10, Añadir Gastos

RSF-11			
Titulo	Gestionar conferencias		
Descripción	El sistema debe permitir al usuario gestionar las conferencias de un proyecto permitiendo las siguientes opciones: <ul style="list-style-type: none"> Consultar conferencias Añadir conferencias (solo jefe del proyecto) 		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-08

Tabla 51: RSF-11, Gestionar conferencias

RSF-12			
Titulo	Consultar conferencias		
Descripción	El sistema debe permitir al usuario visualizar las conferencias del proyecto seleccionado.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-09

Tabla 52: RSF-12, Consultar conferencias

RSF-13			
Titulo	Añadir conferencias		
Descripción	El sistema debe permitir al usuario añadir conferencias al proyecto seleccionado. Para poder añadir una conferencia el usuario debe ser el jefe del proyecto.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-10

Tabla 53: RSF-13, Añadir conferencias

RSF-14			
Titulo	Buscar un proyecto		
Descripción	El sistema debe permitir al usuario buscar proyectos. Este buscador deberá hacer uso de las tecnologías semánticas.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUC-11

Tabla 54: RSF-14, Buscar un proyecto

3.4.2 Requisitos de software no funcionales

A continuación vamos a definir los requisitos de software no funcionales. Los vamos a dividir en dos categorías:

- Requisitos no funcionales de Operación (RSNFO)
- Requisitos no funcionales de interfaz (RSNFI)

3.4.2.1 Requisitos no funcionales de Operación

RSNFO-01			
Titulo	Disponibilidad horaria		
Descripción	La página debe estar disponible las 24 horas del día.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUR-03

Tabla 55: RSNFO-01, Disponibilidad horaria

RSNFO-02			
Titulo	Disponibilidad en varios navegadores		
Descripción	El sistema debe estar disponible en los principales navegadores web.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUR-02

Tabla 56: RSNFO-02, Disponibilidad en varios navegadores

3.4.2.2 Requisitos no funcionales de interfaz

RSNFI-01			
Titulo	Idioma		
Descripción	El idioma de la aplicación debe ser el castellano.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUR-01

Tabla 57: RSNFI-01, Idioma

RSNFI-02			
Titulo	Interfaz sencilla		
Descripción	La interfaz debe facilitar la navegación del usuario en la aplicación.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUR-04

Tabla 58: RSNFI-02, Interfaz sencilla

RSNFO-03			
Titulo	Resolución de la interfaz		
Descripción	La interfaz debe adaptarse a los diferentes tipos de pantalla.		
Prioridad	Alta	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Necesidad	Alta	Fuente	RUR-04

Tabla 59: RSNFI-03, Resolución de la interfaz

4 DISEÑO

En este capítulo se van a exponer las fases de diseño e implementación de la solución escogida. Se va a presentar la arquitectura del sistema, las herramientas que se van a usar para su desarrollo y por ultimo un prototipo para nuestra propuesta de solución.

4.1 Arquitectura del sistema

En este apartado se va a presentar patrón de arquitectura que van a seguir los componentes de nuestra solución.

Para este proyecto se vamos a utilizar la arquitectura **modelo-vista-controlador** (MVC). Esta arquitectura nos permite dividir el sistema en tres capas donde, tenemos la encapsulación de los datos, la interfaz o vista y por último la lógica interna o controlador. El uso de este modelo, facilita el acceso a los diferentes componentes, lo que resulta de gran utilidad si en algún momento estos requieren alguna modificación o arreglo en el futuro, garantizando así el correcto funcionamiento del resto.

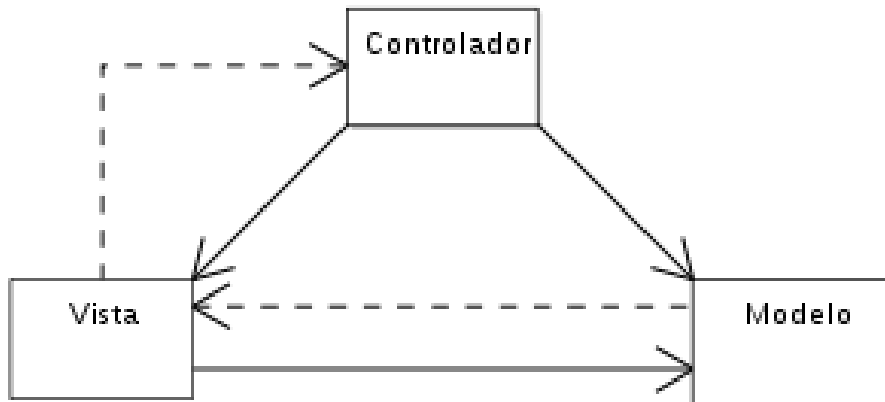


Ilustración 18: MVC

- El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El modelo es el responsable:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.

En nuestro caso el modelo se realizará mediante una capa DAO (Data Access Object), que es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos.

Una de las ventajas de usar DAO, es que al cualquier objeto de negocio no requiere conocimiento directo del destino final de la información que manipula. Los DAO pueden usarse en java para aislar una aplicación de la API de persistencia de java. Por lo que al usar DAO podemos actualizar o cambiar el api de persistencia de java, sin cambiar otras partes de la aplicación, cosa que puede ser muy útil en modificaciones futuras de la aplicación.

Se han utilizado los siguientes schemas:

```
{  
  
  id_proyecto: Number,  
  título: String,  
  descripcion: String,  
  fecha_inicio: Date,  
  fecha_fin: Date,  
  presupuesto: Number,  
  Jefe_proyecto: { id_investigador: Number, dni: String, nombre: String, apellido: String,  
    fecha_nacimiento: Date, departamento: String },  
  investigadores: [{ id_investigador: Number, dni: String, nombre: String, apellido: String,  
    fecha_nacimiento: Date, departamento: String }],  
  
  gastos: [{ id_gasto: Number, tipo: String, cantidad: Number, fecha_gasto: Date,  
    id_investigador: Number }],  
  
  conferencias: [{ id_conferencia: Number, titulo: String, fecha_conferencia: Date, lugar: String,  
    investigadores: [{id_investigador: Number, dni: String, nombre: String, apellido: String}]}]  
}
```

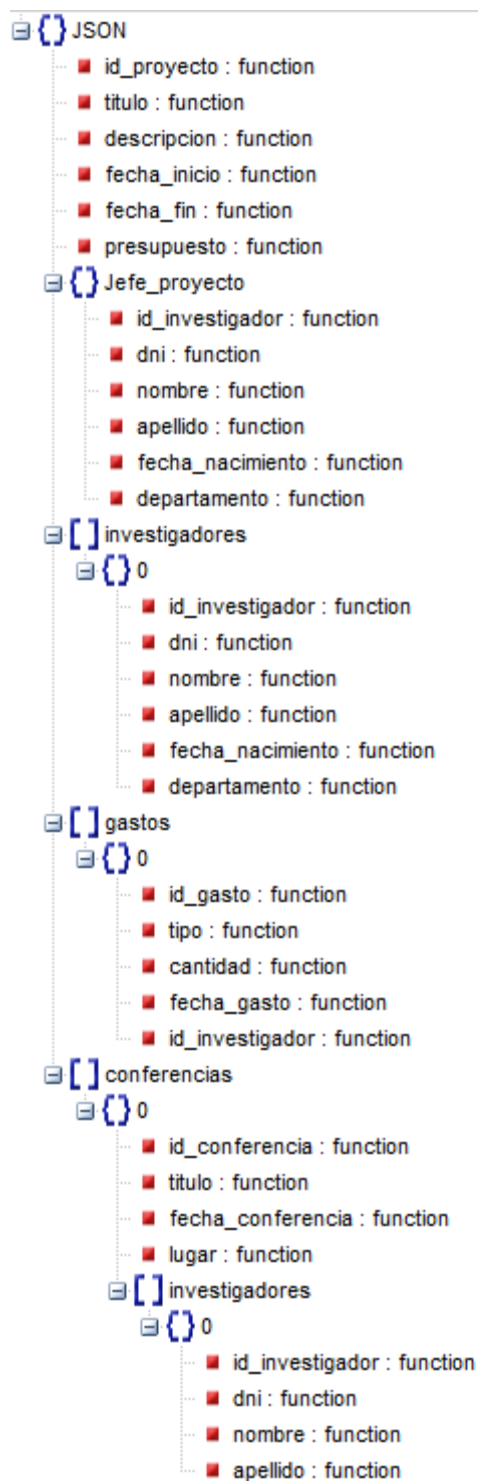


Ilustración 19: Schema Proyecto

```
{  
  id_investigador: Number,  
  proyectos: [{ id_proyecto: Number}]  
}
```

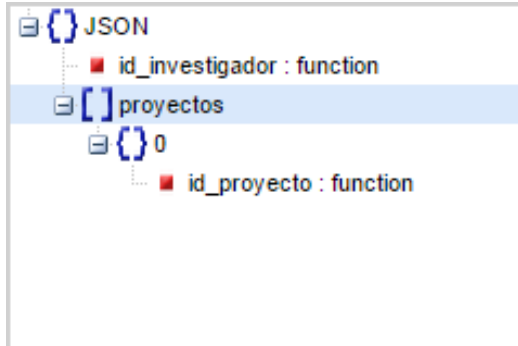


Ilustración 20: Schema Investigador-Proyectos

El controlador es responsable:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega(nueva_orden_de_venta)".

En nuestro caso el controlador será implementado con spring, haciendo uso de un servicio RESTfull para atender a las peticiones enviadas desde la vista.

Las vistas son responsables:


- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

4.2 Interfaz web

En este apartado se va a exponer la interfaz de la web.

Para no sobrecargar la memoria, con capturas de pantalla de la aplicación solo se mostraran las pantallas con entradas de datos en la aplicación.

4.2.1 Login



The image shows a login form titled "P.G.G.I." centered on a blue background. The form has a dashed line separator at the top. Below the title, there are two input fields: "Usuario" and "Contraseña". The "Usuario" field is a single-line text input. The "Contraseña" field is a single-line text input with a small eye icon on the right side to toggle visibility. Below the password field is a blue button labeled "Iniciar sesión".

Ilustración 21: Login

4.2.2 Formularios

4.2.2.1 Formulario dar de alta un investigador

Formulario para dar de alta un investigador. El formulario está titulado "DAR DE ALTA UN INVESTIGADOR" y contiene los siguientes campos de entrada:

- DNI**: Campo de texto con el placeholder "DNI".
- Nombre**: Campo de texto con el placeholder "Nombre".
- Apellidos**: Campo de texto con el placeholder "Apellidos".
- Fecha de Nacimiento**: Campo de texto con el placeholder "Fecha de Nacimiento".
- Departamento**: Campo de texto con el placeholder "Departamento".

En la parte inferior del formulario hay dos botones: "Dar de Alta" y "Cancelar".

Ilustración 22: Formulario dar de alta un investigador

4.2.2.2 Formulario dar de baja un investigador

Formulario para dar de baja un investigador. El formulario está titulado "DAR DE BAJA UN INVESTIGADOR" y contiene los siguientes campos de entrada:

- DNI**: Campo de texto con el placeholder "DNI".
- Departamento**: Campo de texto con el placeholder "Departamento".

En la parte inferior del formulario hay dos botones: "Dar de Baja" y "Cancelar".

Ilustración 23: Formulario dar de baja un investigador

4.2.2.3 Formulario dar de Alta un proyecto

DAR DE ALTA UN PROYECTO

Título

Descripción

Fecha de inicio


Fecha de Finalización

Presupuesto

Jefe de Proyecto

Ilustración 24: Formulario dar de Alta un proyecto

4.2.2.4 Formulario añadir investigador



Formulario para añadir un investigador. El formulario está titulado "AÑADIR INVESTIGADOR" y contiene los siguientes campos de entrada:

- DNI**: Campo de texto con el placeholder "DNI".
- Nombre**: Campo de texto con el placeholder "Nombre".
- Apellidos**: Campo de texto con el placeholder "Apellidos".
- Departamento**: Campo de texto con el placeholder "Departamento".

En la parte inferior del formulario hay dos botones: "Añadir" y "Cancelar".

Ilustración 25: Formulario añadir investigador

4.2.2.4 Formulario añadir gasto




Formulario para añadir un gasto. El formulario está titulado "AÑADIR GASTO" y contiene los siguientes campos de entrada:

- Concepto**: Campo de texto con el placeholder "Concepto".
- Cantidad**: Campo de texto con el placeholder "€".
- Fecha del gasto**: Campo de texto con el placeholder "Fecha del gasto".
- Investigador**: Campo de texto con el placeholder "Investigador".

En la parte inferior del formulario hay dos botones: "Añadir" y "Cancelar".

Ilustración 26: Formulario añadir gasto

4.2.2.5 Formulario añadir conferencia



AÑADIR CONFERENCIA

Título

Lugar

Fecha de la conferencia

Investigador

Ilustración 27: Formulario añadir conferencia

4.2.1 Buscador

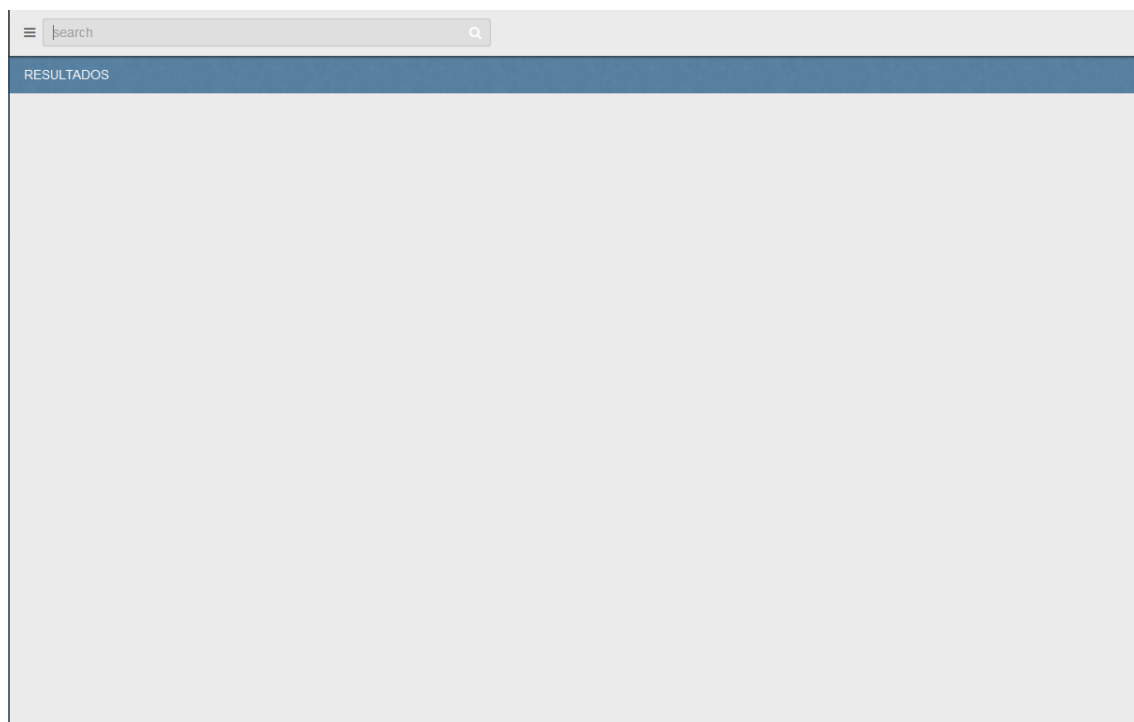


Ilustración 28: Buscador

5 PRUEBAS

Se conoce con el nombre de pruebas software, a una serie de procesos de evaluación del producto software, con el objetivo de proporcionar una información sobre la calidad de este. Estas pruebas son una actividad que forma parte del proceso de calidad del software.

En este apartado vamos a especificar un plan de pruebas de aceptación, para verificar los requisitos establecidos. Al superar todas las pruebas podremos considerar que nuestro proyecto ha finalizado con éxito, cumpliendo así con los objetivos marcados inicialmente.

5.1 Pruebas de aceptación

Las pruebas de aceptación seguirán el siguiente patrón:

Identificador ¹	
Título ²	
Entrada ³	
Salida ⁴	
Dependencia ⁵	

Tabla 60: Patrón Pruebas

1. **Identificador:** Código que identifica de forma unívoca cada prueba. Seguirá el siguiente formato: PA-XX, donde XX es un número correlativo incremental.
2. **Título:** Título descriptivo de la prueba.
3. **Entrada esperada:** Acción de entrada o dato que debe ser proporcionado para realizar la prueba.
4. **Salida esperada:** Resultado esperado al realizarse la prueba.
5. **Dependencia:** Dependencia entre los casos de uso y la prueba.

PA-01	
Titulo	Dar de alta un investigador
Entrada	El usuario administrador accede al formulario de registro de un investigador.
Salida	Se muestra el formulario de registro. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Enviar”, se registra un investigador.
Dependencia	CUA-01

Tabla 61: PA-01, Dar de alta un investigador

PA-02	
Titulo	Dar de baja un investigador
Entrada	El usuario administrador accede al formulario para dar de baja un investigador.
Salida	Se muestra el formulario para dar de baja. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Enviar”, se da de baja un investigador.
Dependencia	CUA-02

Tabla 62: PA-02, Dar de baja un investigador

PA-03	
Titulo	Dar de alta un proyecto
Entrada	El usuario administrador accede al formulario de registro de un proyecto.
Salida	Se muestra el formulario de registro. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Enviar”, se da alta un proyecto.
Dependencia	CUA-03

Tabla 63: PA-03, Dar de alta un proyecto

PA-04	
Titulo	Visualizar proyectos usuario investigador
Entrada	El usuario investigador accede a la web con los credenciales de usuario investigador y pulsa en la opción “visualizar proyectos”.
Salida	Se muestran los proyectos en los que ese usuario investigador esta dado de alta.
Dependencia	CUI-01

Tabla 64: PA-04, Visualizar proyectos usuario investigador

PA-05	
Titulo	Visualizar proyectos usuario jefe proyecto
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe de proyecto y pulsa en la opción “visualizar proyectos”.
Salida	Se muestran los proyectos en los que ese usuario jefe proyecto está dado de alta.
Dependencia	CUJ-01

Tabla 65: PA-05, Visualizar proyectos usuario jefe proyecto

PA-06	
Titulo	Ver información del proyecto usuario investigador.
Entrada	El usuario investigador accede a la web con los credenciales de usuario investigador y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Ver información del proyecto”.
Salida	Se muestra la información del proyecto seleccionado.
Dependencia	CUI-02

Tabla 66: PA-06, información del proyecto usuario investigador.

PA-07	
Titulo	Ver información del proyecto usuario jefe proyecto.
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe proyecto y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Ver información del proyecto”.
Salida	Se muestra la información del proyecto seleccionado.
Dependencia	CUJ-02

Tabla 67: PA-07, Ver información del proyecto usuario jefe proyecto.

PA-08	
Titulo	Visualizar proyectos usuario investigador
Entrada	El usuario administrador accede a la web con los credenciales de usuario investigador y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Añadir investigador”.
Salida	Se muestra el formulario para añadir un investigador a ese proyecto. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Enviar”, se da alta un investigador en ese proyecto.
Dependencia	CUJ-03

Tabla 68: PA-08, Visualizar proyectos usuario investigador

PA-09	
Titulo	Visualizar gastos usuario investigador
Entrada	El usuario investigador accede a la web con los credenciales de usuario investigador y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar gastos”.
Salida	Se muestra el menú para consultar o añadir gastos sobre el proyecto seleccionado.
Dependencia	CUI-03

Tabla 69: PA-09, Visualizar gastos usuario investigador

PA-10	
Titulo	Visualizar gastos usuario jefe proyecto.
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe proyecto y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar gastos”.
Salida	Se muestra el menú para consultar o añadir gastos sobre el proyecto seleccionado.
Dependencia	CUJ-04

Tabla 70: PA-10, Visualizar gastos usuario jefe proyecto.

PA-11	
Titulo	Consultar gastos usuario investigador
Entrada	El usuario investigador accede a la web con los credenciales de usuario investigador y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar gastos”. En el siguiente menú selecciona la opción “Consular gastos”.
Salida	Se muestra la información de los gastos del proyecto seleccionado.
Dependencia	CUI-04

Tabla 71: PA-11, Consultar gastos usuario investigador

PA-12	
Titulo	Consultar gastos usuario jefe proyecto.
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe proyecto y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar gastos”. En el siguiente menú selecciona la opción “Consular gastos”.
Salida	Se muestra la información de los gastos del proyecto seleccionado.
Dependencia	CUJ-05

Tabla 72: PA-12, Consultar gastos usuario jefe proyecto.

PA-13	
Titulo	Añadir gastos usuario investigador
Entrada	El usuario investigador accede a la web con los credenciales de usuario investigador y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar gastos”. En el siguiente menú selecciona la opción “Añadir gastos”.
Salida	Se muestra el formulario para añadir un gasto a ese proyecto. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Enviar”, se da alta un gasto en ese proyecto.
Dependencia	CUI-05

Tabla 73: PA-13, Añadir gastos usuario investigador

PA-14	
Titulo	Añadir gastos usuario jefe proyecto.
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe proyecto y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar gastos”. En el siguiente menú selecciona la opción “Añadir gastos”.
Salida	Se muestra el formulario para añadir un gasto a ese proyecto. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Enviar”, se da alta un gasto en ese proyecto.
Dependencia	CUJ-06

Tabla 74: PA-14, Añadir gastos usuario jefe proyecto.

PA-15	
Titulo	Visualizar conferencias usuario investigador
Entrada	El usuario investigador accede a la web con los credenciales de usuario investigador y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar conferencias”.
Salida	Se muestra el menú para consultar conferencias sobre el proyecto seleccionado.
Dependencia	CUI-06

Tabla 75: PA-15, Visualizar conferencias usuario investigador

PA-16	
Titulo	Visualizar conferencias usuario jefe proyecto.
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe proyecto y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar conferencias”.
Salida	Se muestra el menú para consultar o añadir conferencias sobre el proyecto seleccionado.
Dependencia	CUJ-07

Tabla 76: PA-16, Visualizar conferencias usuario jefe proyecto.

PA-17	
Titulo	Consultar conferencias usuario investigador
Entrada	El usuario investigador accede a la web con los credenciales de usuario investigador y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar conferencias”. En el siguiente menú selecciona la opción “Consular conferencias”.
Salida	Se muestra la información de los conferencias del proyecto seleccionado.
Dependencia	CUI-07

Tabla 77: PA-17, Consultar conferencias usuario investigador

PA-18	
Titulo	Consultar conferencias usuario jefe proyecto.
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe proyecto y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar conferencias”. En el siguiente menú selecciona la opción “Consular conferencias”.
Salida	Se muestra la información de los conferencias del proyecto seleccionado.
Dependencia	CUJ-08

Tabla 78: PA-18, Consultar conferencias usuario jefe proyecto.

PA-19	
Titulo	Añadir conferencias usuario jefe proyecto.
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe proyecto y pulsa en la opción “visualizar proyectos”. Después pulsa la opción “Visualizar conferencias”. En el siguiente menú selecciona la opción “Añadir conferencias”.
Salida	Se muestra el formulario para añadir una conferencia a ese proyecto. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Enviar”, se da alta una conferencia en ese proyecto.
Dependencia	CUJ-09

Tabla 79: PA-19, Añadir conferencias usuario jefe proyecto.

PA-20	
Titulo	Buscar proyectos usuario investigador
Entrada	El usuario investigador accede a la web con los credenciales de usuario investigador y pulsa en la opción “Buscar proyectos”.
Salida	Se muestra el formulario para añadir buscar un proyecto. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Buscar” se muestran los resultados obtenidos.
Dependencia	CUI-08

Tabla 80: PA-20, Buscar proyectos usuario investigador

PA-21	
Titulo	Buscar proyectos usuario jefe proyecto
Entrada	El usuario jefe proyecto accede a la web con los credenciales de usuario jefe de proyecto y pulsa en la opción “Buscar proyectos”.
Salida	Se muestra el formulario para añadir buscar un proyecto. El usuario debe rellenarlo en función de unos valores predeterminados y con el formato exigido. Después de introducir todos los datos correctamente y pulsar el botón “Buscar” se muestran los resultados obtenidos.
Dependencia	CUJ-10

Tabla 81: PA-21, Buscar proyectos usuario jefe proyecto

Matriz de trazabilidad entre los casos de uso y los casos de prueba que los cubren

5.2 Resultados de las pruebas

Tras finalizar la ejecución de las pruebas, procedemos a evaluar si el resultado de la salida de las mismas es satisfactorio.

Identificador	Resultado de la salida
PA-01	Correcto
PA-02	Correcto
PA-03	Correcto
PA-04	Correcto
PA-05	Correcto
PA-06	Correcto
PA-07	Correcto
PA-08	Correcto
PA-09	Correcto
PA-10	Correcto
PA-11	Correcto
PA-12	Correcto
PA-13	Correcto
PA-14	Correcto
PA-15	Correcto
PA-16	Correcto
PA-17	Correcto
PA-18	Correcto
PA-19	Correcto
PA-20	Correcto
PA-21	Correcto

Tabla 82. Resultados pruebas

El resultado de las pruebas ha sido satisfactorio, puesto que se han obtenido las salidas esperadas en todas y cada una de las pruebas planificadas.

6 GESTION DEL PROYECTO

Esta capitulo describe la gestión realizada en el proyecto. Se divide en dos apartados:

- Planificación del Proyecto
- Análisis Económico

6.1 Planificación del proyecto

En esta apartado se describen todas las tareas que se han realizado durante el desarrollo de este proyecto.

En el siguiente diagrama de Gantt se puede observar las tareas del proyecto, así como su planificación.

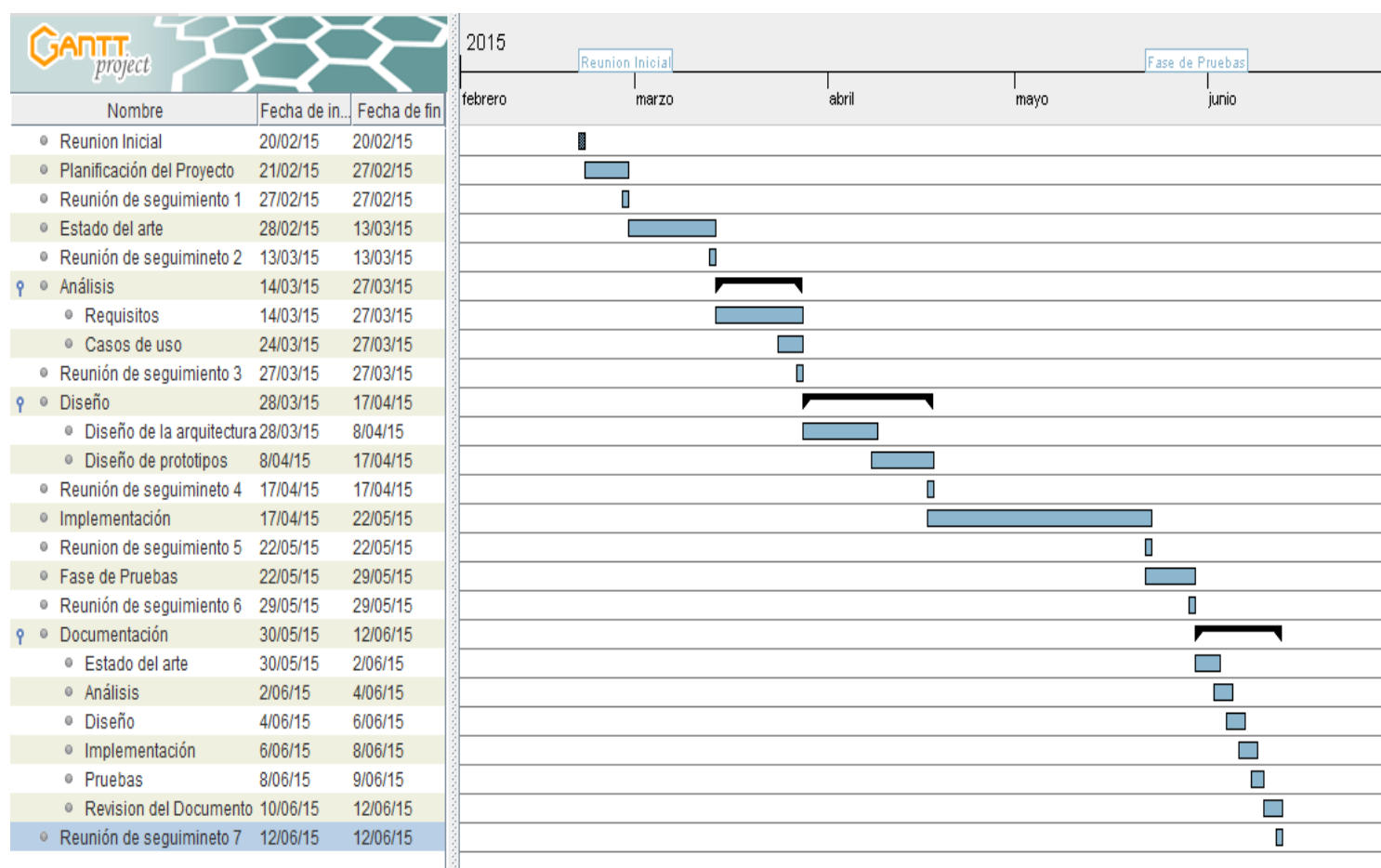


Ilustración 29: Diagrama Gantt

6.2 Análisis económico

A continuación una vez determinado el alcance de las tareas del proyecto, se va a proceder a detallar el coste del mismo.

Vamos a desglosar los gastos en los siguientes apartados:

- Gastos de personal.
- Gastos Hardware.
- Gastos Software.
- Coste total del proyecto.

6.2.1 Gastos de personal

A continuación vamos a computar los gastos generados por el personal encargado de la implementación de la aplicación.

Como personal se ha incluido a las siguientes personas:

- Enrique Colen Lozano: Desarrollador de la aplicación.
- Juan Miguel Gómez Berbis: Tutor del proyecto.

Nombre	Categoría	Importe hora (€)	Horas a facturar	Importe (€)
Enrique Colen Lozano	Ingeniero Junior	29 €	320	9.280 €
Juan Miguel Gómez Berbis	Jefe de proyecto	46 €	30	1.380 €
Total		75 €		10.660 €

Tabla 83: gastos personal

Para la realización del cálculo de las horas imputadas por el ingeniero (en este caso el alumno), se ha establecido que a lo largo del tiempo de realización del proyecto, se han realizado 20 horas semanales.

Los costes imputados incluyen los valores del **IRPF** y **SS**.

6.2.2 Gastos Hardware

A continuación se desglosan los gastos del hardware empleados para el desarrollo del proyecto.

Concepto	Unidades	Importe
Portátil Lenovo	1	800 €
Bq aquaris E4.5	1	150 €
Total		950 €

Tabla 84: Gastos hardware

6.2.3 Gastos Software

A continuación se desglosan los gastos del hardware empleados para el desarrollo del proyecto.

Todas las herramientas utilizadas durante el desarrollo del proyecto no son de pago, lo cual influye positivamente en el coste económico de nuestro proyecto

Concepto	Unidades	Importe
GitHub	1	0 €
ganttProject	1	0 €
Eclipse	1	0 €
MongoDb	1	0 €
Windows 7	1	0 €
Total		0 €

Tabla 85: Gastos software

- El coste de la licencia de Windows es gratuito, ya que la universidad nos proporciona una licencia por ser alumnos de la universidad.

6.2.4 Coste total del proyecto

Concepto	Coste
Gastos de personal	10.660 €
Gastos Hardware Empleado	950 €
Gastos Software Empleado	0 E
Total	11.610 €

Tabla 86: Coste total proyecto

El coste total del desarrollo del proyecto es ONCE MIL SEISCIENTOS DIEZ EUROS (**11.610 €**) IVA no incluido.

7 CONCLUSIONES

En esta apartado se exponen las conclusiones obtenidas tras haber finalizado el proyecto.

Se van a dividir en dos apartados. El primer apartado son las conclusiones relacionadas con la realización del proyecto, mientras que el segundo apartado son las líneas futuras para las posibles mejoras de la aplicación.

7.1 Conclusiones del proyecto

Con el desarrollo de este proyecto se ha obtenido una aplicación web capaz de gestionar grupos de investigación que hace uso de tecnologías semánticas. Esta aplicación se puede visualizar en cualquier dispositivo ya que hemos usado un diseño responsivo.

Otro factor a favor es la arquitectura utilizada, ya que esta, nos permite adaptarnos fácilmente a los posibles cambios que se realicen en la aplicación en un futuro. Por ejemplo, cambiar el sistema de base de datos o las tecnologías web utilizadas.

Durante el desarrollo del proyecto hemos podido estudiar, analizar y aplicar las tecnologías semánticas en el marco de nuestro proyecto. Cosa bastante útil ya que esto puede resultar un elemento diferenciador con aplicaciones similares.

En lo que se refiere a las tecnologías utilizadas, hemos podido desarrollar con tecnologías punteras como Spring o MongoDB. Esto puede ser un factor importante de cara al mercado laboral del desarrollador, ya que las empresas hoy en día buscan desarrolladores que conozcan estas tecnologías.

Por ultimo cabe destacar que el proyecto ha sido iniciado desde cero. Pasando así por todas las fases del desarrollo, poniendo en práctica conocimientos adquiridos a lo largo de la carrera.

7.2 Líneas futuras

En este apartado se van a describir las líneas futuras que puede tomar el proyecto. Estas líneas se basan en las posibles vías de mejora que tiene el proyecto para poder tener una mejora en su aceptación en el mercado.

Una posible línea de mejora es hacer que sea una plataforma multi-idioma. En España además del castellano, existen más lenguas cooficiales según la comunidad autónoma. Además para poder ser una aplicación que se pueda utilizar no solo en el ámbito nacional, sería positivo que la plataforma diera soporte a idiomas como el Inglés, Francés o Alemán.

Otra posible línea de mejora, puede ser la integración de esta plataforma con los repositorios de los proyectos, tanto a nivel de documentación como de código. Esto puede ser una tarea algo difícil, puesto que no todos los grupos de investigación utilizan los mismos.

Otra vía para mejorar el rendimiento de nuestra aplicación, sería clasificar los investigadores según sus áreas de trabajo y especialidades, para que una vez que estos terminen un proyecto, puedan ser asignados de forma eficiente a otros proyectos acordes con sus capacidades. Esta posible mejora se podría realizar usando tecnologías semánticas.

Otra parte importante para mejorar el proyecto, es el feedback que pueden proporcionar los usuarios que hagan de la aplicación. Hacer un buen uso de este feedback puede ayudarnos bastante, ya que podemos mejorar la funcionalidad de nuestra aplicación acorde con las necesidades que le van surgiendo al usuario de la misma.

Por último, con el transcurso del tiempo sería bueno realizar un estudio de nuevas tecnologías. La web evoluciona de forma vertiginosa, aunque las grandes empresas suelen tardar en adaptarse a las nuevas tecnologías que van surgiendo, las pequeñas y medianas suelen aprovechar estas oportunidades para obtener una posición de ventaja en el mercado. Por ejemplo, podrían surgir nuevas tecnologías de desarrollo web o de almacenamiento de base datos, que podrían reemplazar a las utilizadas durante el desarrollo de este proyecto.

8 REFERENCIAS y BIBLIOGRAFÍA

1. Guía breve de la web semántica:
<http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>
2. Actividad de Web Semántica: <http://www.w3.org/2001/sw/>
3. Visión General del Lenguaje de Ontologías Web: <http://www.w3.org/TR/owl-features/>
4. Página principal de RDF: <http://www.w3.org/RDF/>
5. SPARQL: http://www.w3.org/2009/sparql/wiki/Main_Page
6. Página principal de OWL: <http://www.w3.org/2001/sw/wiki/OWL>
7. Estadísticas YouTube: <https://www.youtube.com/yt/press/es/statistics.html>
8. Estadísticas redes sociales: <http://vilmanunez.com/2015/04/27/estadisticas-de-redes-sociales-2015-convencer-a-tus-clientes/>
9. Grupos de investigación: <http://profesores.universia.es/investigacion/grupos-investigacion/>
10. Madri+d: <http://www.madrimasd.org/Investigadores/buscador-grupos-investigacion/default.asp>
11. CSIC: <http://www.csic.es/>
12. Agile Alliance: <http://www.agilealliance.org/>
13. Metodologías ágiles: http://es.wikipedia.org/wiki/Manifiesto_%C3%A1gil
14. Metodologías ágiles
15. Canós, J. H., Letelier, P., & Penadés, M. C. (2003). Metodologías Ágiles en el desarrollo de
16. Software. *VIII Jornadas de Ingeniería de Software y Bases de Datos, JISBD*.
17. RUP
18. Martínez, A., & Martínez, R. (2002). Guía a Rational Unified Process.
19. Bootstrap: <http://getbootstrap.com/>
20. Diseño web responsivo <http://www.esandra.com/disenio-web-responsive-usar-framework>
21. MongoDB <https://www.mongodb.org/>
22. Spring <https://spring.io/>
23. Apache Marmotta <http://marmotta.apache.org/>
24. GitHub: <https://netbeans.org/>
25. GanttProject: <http://www.ganttproject.biz/>
26. Java Enterprise Edition:
27. Groussard, T. (2010). Java Enterprise Edition: Desarrollo de aplicaciones web con JEE 6.